UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería en Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Estudio e implementación de algoritmos para detección de anomalías en entornos de videovigilancia

José Gil Torrecilla
Tutor: Luis Salgado Álvarez de Sotomayor
Ponente: Jesús Bescós Cano

Julio 2017

Estudio e implementación de algoritmos para detección de anomalías en entornos de videovigilancia

AUTOR: José Gil Torrecilla
TUTOR: Luis Salgado Álvarez de Sotomayor

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2017

Resumen (castellano)

Este Trabajo Fin de Grado tiene como objetivo el estudio y desarrollo de distintas técnicas de análisis de secuencias de videovigilancia que permitan la detección de anomalías presentes en las mismas para, a partir de ello, construir un sistema completo que permita llevar a cabo esta tarea.

Basándose en la literatura y otros trabajos sobre detección de anomalías en vídeo, se ha diseñado el detector, compuesto por diferentes etapas de procesamiento que se han ido analizando de manera independiente. De este diseño, donde más se ha profundizado ha sido en la elección y extracción de características, decidiéndose por las características espacio-temporales de los gradientes de intensidad para este trabajo.

A partir de estas, se ha establecido un método de entrenamiento y aprendizaje y se han implementado y analizado dos algoritmos para modelar el comportamiento de las escenas, evolucionando desde un sistema basado en distancias para detectar las anomalías hasta uno basado en niveles de pertenencia a cada clase, que son *K-Means* y *Fuzzy C-Means*. Se analizan las ventajas e inconvenientes de cada uno, y se encuentra el ajuste óptimo de ambos a través de una serie de pruebas predefinidas para ello.

El estudio de los resultados se realiza a través de secuencias de videovigilancia reales en diferentes entornos y situaciones, para todas las distintas secuencias utilizadas y para cada uno de los modelos propuestos, siguiendo de esta manera una configuración de lo particular a lo general motivado por la poca similitud existente entre cada una de las secuencias.

Para concluir el trabajo, se verifica si se han alcanzado los objetivos marcados y se trata de extraer conclusiones sobre las características empleadas y los algoritmos implementados, así como establecer una serie de posibles mejoras futuras para el sistema.

Palabras clave (castellano)

Anomalías, videovigilancia, visión artificial, gradientes, inteligencia artificial, escaleras mecánicas, seguridad, clasificación.

Abstract (English)

This Bachelor Thesis aims to study and develop different techniques for de analysis of video sequences in a surveillance environment in order to detect anomalies present in the scene, and therefore build a complete system to carry out this task.

Based on the literature and other papers about anomaly detection in video, the detector has been designed from blocks, and every block has been analyzed separately. In this design, there has been deepened specially in the feature choice and extraction, being decided in this thesis for the spatio-temporal features of the intensity gradients.

From these, it has been stablished a training and learning method, and two algorithms for modeling the scene behavior have been implemented and analyzed, evolving from a distance based system to detect anomalies until one based on the amount of membership to each class which are *K-Means* and *Fuzzy C-Means*. The advantages and disadvantages of each method are analyzed and on adequate adjustment is found throughout a series of predefined tests.

The analysis of results is done with real video surveillance sequences from different environments and situations, for every sequence and every model proposed, thus following a configuration from particularity to generality, motivated by the small similarity between each of the sequences.

In order to finish the thesis, it is verified if the marked objectives have been reached and it is tried to extract conclusions about the features and algorithms used and implemented, as well as propose some possible future improvements for the system.

Keywords (English)

Anomalies, video-surveillance, computer vision, gradients, artificial intelligence, mechanic stairs, security, clustering.

Agradecimientos

A mis padres, por educarme, cuidarme cada día y confiar en mí; a mi hermano, por todo ello y más; y al resto de mi familia.

A mis amigos de siempre, que nunca me fallan y que nunca me falten, y a los nuevos que aquí he conocido.

A ti, aunque te moleste, por aguantarme, que no es poco, y hacerme tan feliz.

A los buenos profesores que he tenido, por transmitirme no sólo conocimiento sino actitud y ganas de adquirirlo.

A todas y cada una de las personas que han participado en mi vida y me han hecho llegar hasta aquí, y a aquellas que a partir de ahora vengan.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

La detección de eventos anómalos en entornos de videovigilancia es una tarea que ha sido muy estudiada por la comunidad científica en los últimos años pero en la que, sin embargo, aún no se ha alcanzado un estado de madurez y permanece por tanto abierta. Por ello, y por la enorme cantidad de posibles aplicaciones de gran importancia para la seguridad que puede cobrar un sistema que sea capaz de detectar con precisión este tipo de eventos, se ha decidido realizar este trabajo.

Se va a elaborar por tanto en este documento un sistema completo que detecte eventos anómalos en escenarios públicos, concretamente en escenas de interiores de escaleras mecánicas, mediante técnicas de visión artificial.

1.2 Objetivos

El objetivo de este TFG es construir desde cero un sistema detector de anomalías basado en el modelado de la normalidad de la escena y analizar su funcionamiento. Lo más relevante y destacable del mismo es que tanto el diseño del mismo como su implementación se realizan íntegramente en este trabajo sin tomar ningún otro como base, es decir, no se mejora, modifica o reutiliza un sistema más complejo, sino que se crea uno por completo que, aunque más sencillo y simplificado, sea plenamente funcional.

Para su realización, se llevará a cabo como primer paso un estudio del estado del arte en detección de anomalías para ver qué métodos y características se emplean en la literatura de esta disciplina y cuáles de ellos pueden encajar mejor en el escenario particular de este trabajo. Con ello, se tomarán aquellas ideas que se consideren apropiadas y se diseñará e implementará el sistema detector de anomalías.

Para terminar, se analizará y ajustará el sistema propuesto mediante bases de datos públicas creadas para la detección de anomalías y mediante secuencias de escaleras mecánicas que se deberán etiquetar manualmente para construir unos datos de referencia (*Ground-Truth*), y se evaluará el éxito o no del mismo.

1.3 Organización de la memoria

Esta memoria consta de los siguientes capítulos:

- Estado del arte. Se hará una revisión de los procedimientos, métodos y bases teóricas de los distintos sistemas existentes en la actualidad para detección de anomalías en entornos de videovigilancia, haciendo especial hincapié en los que se consideren más relevantes para este trabajo.
- Diseño del sistema. Se presentará y explicará la implementación particular de las distintas técnicas vistas en el estado del arte elegidas para el detector que aquí se propone. Esto se hará siguiendo una estructura funcional de bloques en la que se comenzará por una vista general del sistema para ir detallando cada uno de los sub-sistemas que lo conforman.

- Pruebas y análisis de resultados. Se ejecutará el detector con distintas secuencias de prueba para demostrar el correcto funcionamiento del mismo. Primero se pasará una secuencia de datos pública y muy trabajada en la literatura, para comparar el rendimiento del sistema implementado con otros existentes, y posteriormente se analizarán secuencias de escaleras mecánicas, pues son el objetivo final del trabajo.
- Conclusiones y trabajo futuro. En este apartado se tratará de analizar a posteriori cuán bueno ha sido el funcionamiento del detector implementado, en qué grado se han cumplido las expectativas previstas y por qué esto ha sido así. Además, se propondrán una serie de posibles líneas de mejora para este trabajo en el caso de que en un futuro se desee continuar con el mismo o tomarlo como base para otros.
- Anexos. Se emplearán para documentar debidamente las bases teóricas y matemáticas necesarias para la completa comprensión del sistema diseñado, así como el conjunto de los resultados obtenidos en las pruebas de validación. Servirán, por tanto, como apoyo para las partes de estado del arte y de diseño del sistema.

2 Estado del arte

2.1 Introducción

El objetivo final de este trabajo es la detección de anomalías dentro de la disciplina de la videovigilancia y del reconocimiento de eventos, campo sobre el que, si bien existen multitud de publicaciones, sigue siendo objeto de estudio en la actualidad. Como es lógico, y dada la disponibilidad de literatura al respecto, antes de abordar directamente el problema se hará un análisis previo para conocer las diferentes propuestas y su funcionamiento.

En toda la bibliografía estudiada se hacen distintas clasificaciones y divisiones a la hora de explicar los detectores propuestos, a pesar de que finalmente todos tienen una estructura de bloques similar. En este trabajo se han utilizado como base las publicaciones [1, 2], y se parte de la clasificación de bloques utilizada en [1] por no estar únicamente centrada en multitudes como [2] y establecer un diagrama de flujo desde la señal de entrada hasta la decisión final. No obstante, se modifica levemente esta clasificación de acuerdo a una configuración que se considera más adaptada a este trabajo, y de esta manera se tratarán en este apartado la definición de anomalía y el objetivo de vigilancia, la extracción de características, el método de aprendizaje y el modelado.

A continuación se desglosa de esta manera la información bibliográfica utilizada para este trabajo, y se valoran las posibles virtudes e inconvenientes que puedan encontrarse a la hora de incorporar las ideas que en las publicaciones se mencionan.

2.2 Definición de anomalía

Dado que el objetivo final del trabajo es hacer una correcta detección de anomalías, antes de nada es necesario definir concretamente qué es una anomalía, pues esto dependerá del contexto y aplicación en el que se encuentre el sistema, y los supuestos y definiciones que se tomen afectarán a los métodos empleados para la posterior extracción de características.

Entendiendo comportamiento como las acciones observables realizadas por los agentes (personas u otros objetos presentes en la escena), los comportamientos inusuales captan la atención por ser distintos a los patrones regulares en el contexto. Por otro lado, dado que las anomalías en una escena pueden estar juntas o dispersas, ser puntuales o duraderas e incluso presentar alguna determinada estructura [3], definir la anomalía es una tarea complicada y crítica para el éxito y robustez del detector. Por tanto, el objetivo esencial es modelar el comportamiento normal de la escena para poder detectar e identificar los eventos anómalos. Además, Siguiendo la clasificación indicada al inicio [1], se proponen y aceptan en la literatura tres definiciones fundamentalmente distintas de anomalía:

a) Anomalía como evento infrecuente: Se entiende evento anómalo como aquel que ocurre poco tiempo o en pocas ocasiones frente a la totalidad del tiempo de la escena. Estudios como [4] utilizan esta definición para sus detecciones, teniendo en cuenta que para hacer una correcta clasificación será necesario tener una gran cantidad de datos de ejemplo para generar el modelo, pues todo evento no anómalo debe de estar presente en el entrenamiento. Otra limitación de esta definición es el ruido, que si no se detecta correctamente derivará en falsos positivo como se indica en [1].

b) Anomalía como evento de características distintas. No se tiene en cuenta su representación o frecuencia de aparición, sino que se considera anomalía aquellos eventos que son significativamente distintos a los eventos normales. Estudios como [5] consideran anomalía eventos a gran distancia de los datos usados para modelar la normalidad, mientras que otros como [6] se fijan en características como la duración, posición, trayectoria… para hacer su clasificación. La gran limitación de esta definición, y la que la hace inviable para entornos muy críticos de seguridad, es que, si se planea deliberadamente realizar una acción anómala, será imposible detectarla si ésta trata de imitar la normalidad.

c) Anomalía como evento con un significado específico. Esta propuesta, utilizada en [7], es la más compleja de las vistas hasta ahora, pues define los eventos anómalos *a priori* como acciones o sucesos con un significado especifico, lo que puede relacionarse también con un problema de etiquetado de eventos (ajeno a este trabajo). Utilizando esta opción se podrán detectar eventos anómalos que en b) pasarían desapercibidos y se tendrá un porcentaje de acierto muy alto, pero el rango de eventos que se podrán detectar será reducido, pues sólo se tendrá éxito para aquellos eventos que hayan sido modelados específicamente.

A la vista de las definiciones propuestas, se descarta para este trabajo la primera definición por no contar con la suficiente cantidad de datos de entrenamiento. Si se fuera capaz de modelar todos los posibles eventos anómalos o si se deseara modelar algunos muy concretos se optaría por la segunda o tercera opción, pero como se usarán distintas escenas y se pretende ser lo más robusto posible la definición utilizada para este trabajo será la de anomalía como evento de características distintas.

Además, en la literatura analizada se hace referencia al tipo de sensor utilizado, pero en este trabajo no se entrará a valorarlo pues se asumirán secuencias de entrada de vídeo de espectro visible a color o en escala de grises.

2.3 Objetivo

Se define el objetivo o *target* de la vigilancia como la entidad o entidades sobre las que esta opera [1], es decir, el objetivo serán todos los elementos presentes en la escena que sean susceptibles de generar una anomalía. En la Figura 2-1 se muestran los objetivos más comunes en vigilancia.

Figura 2-1: Diagrama de Venn de la clasificación de objetivos de detección de anomalías en vigilancia [1]

Existen, por supuesto, otros *targets*, pero podrían englobarse dentro de alguno de los indicados o mezclarse varios de los grupos. Ejemplos de aplicaciones en este campo serían controles de calidad o de objetos abandonados [8] (objetos); seguridad en aglomeraciones de personas [9] (multitudes); infracciones, atascos o alertas de seguridad [10] (tráfico); y cuidado de personas o robos en casas [11] (individuos). Sin embargo, a menudo en un proyecto de detección de anomalías se podrán encontrar aplicaciones con varios objetivos, como individuos y tráfico [12], por lo que se buscará en estos casos una robustez en las características extraídas y su procesado para poder describir de la mejor forma todas las posibles anomalías en un análisis único y sin tener que tratar cada grupo del diagrama de Venn (Figura 2-1) por separado.

En este trabajo, puesto que se va a trabajar con secuencias de video en las que los objetivos serán grupos de personas con una densidad muy variable (desde varios individuos en escaleras mecánicas hasta aproximadamente medio centenar en UCSD) se implementarán características propias de la detección en multitudes.

2.4 Extracción de características

Debido a la gran cantidad de contextos con características particulares, y como se explica en [3], la clave para un modelado exitoso está en la extracción de características altamente descriptivas y discriminativas para una representación mínima y compacta. Se deben buscar características que idealmente sean invariantes a cambios de traslación, rotación, iluminación… y estas pueden ser globales o locales y espaciales o temporales, o ambas.

Por su parte, las características basadas en objetos aportan una interpretación semántica directa de comportamiento a nivel de objeto. En la mayoría de las ocasiones se emplean dichas características. como trayectorias [13], por su simplicidad o características de bajo nivel como es flujo óptico [14] por sus buenos resultados.

Como se puede ver, las características mencionadas hacen referencia a la información de movimiento y localización espacial para describir un comportamiento, pues aunque otras características tales como geometría, estructura o color pueden ser útiles, es la información de movimiento la base del análisis de escenas con multitudes. Estas características se clasifican según [2] en características basadas en flujo, espacio-temporales locales y trayectorias.

2.4.1 Características basadas en flujo

Son características extraídas a nivel de píxel, y son sin duda las más empleadas por la literatura estudiada [14, 15] por su capacidad para modelar los eventos independientemente del agente o agentes que los producen, por lo que puede inferirse a priori que serán las características que mejores resultados permitan obtener en cuanto a detección de anomalías. Su gran virtud es que, en entornos de multitudes, las acciones de un único individuo pueden parecer aleatorias, pero con una vista general del flujo de movimiento de la escena pueden detectarse patrones e información de interés. Dentro de la categoría de flujo se puede distinguir flujo óptico, flujo de partículas y flujo en rachas o *streak*, que no se entrarán a estudiar en este trabajo puesto que su complejidad de implementación es muy elevada y aquí se pretende construir un detector más sencillo y estudiar hasta qué punto se puede extraer información válida para modelar la escena con otras características más básicas, que puedan servir de apoyo a éstas sobre las que ya se han implementado gran cantidad de sistemas y se ha demostrado ampliamente su eficacia.

2.4.2 Características espacio-temporales locales

Algunas escenas de multitudes, aunque tengan una densidad de individuos similar otras, presentan una variabilidad mayor y están menos estructuradas por lo que el movimiento en distintas áreas puede no ser uniforme e incluso una representación muy fina como el flujo no será capaz de arrojar suficiente información [2]. Para solucionar esto se modelan las relaciones espacio temporales existentes en la escena usando cuadros bidimensionales o cubos tridimensionales, *cuboides*, para calcular gradientes y funciones de histograma.

Los gradientes 3D de cada píxel, estudiados en detalle en [16], representan colectivamente el patrón de movimiento característico en cada una de los cuadros o cuboides, y los histogramas generados a partir de ellos calculan las orientaciones presentes y permiten asignarles pesos. Estas operaciones son costosas y propensas a errores, por lo que se proponen soluciones que traten de solventar estas limitaciones como ODF (Orientation Distribution Function) [17], que no tiene en cuenta la información de magnitud para disminuir el cálculo necesario y por tanto el tiempo de ejecución.

2.4.3 Trayectorias

Dado que normalmente en las escenas el movimiento de una multitud es regular y repetitivo, se pueden analizar las actividades basándose en las características de trayectoria del movimiento de los agentes, teniendo en cuenta la distancia relativa entre ellos, la aceleración o la energía del movimiento [2]. Estudios como [13] basan su funcionamiento en esta idea.

Comparando con las representaciones anteriores, el análisis de trayectorias es más semántico de alto nivel, pero presenta problemas para la detección y seguimiento precisos en multitudes cuando su densidad aumenta. Una opción propuesta para tratar de solucionar este problema consiste en dividir estas trayectorias en fragmentos o *tracklets* de menor longitud [18].

En este trabajo se van a utilizar, tomando como punto de partida el análisis en multitudes, características espacio-temporales basadas en el gradiente por la mayor simplicidad en su implementación y compromiso de resultados para todo tipo de escenas de interiores o exteriores y con densidad de personas muy variable por ser características de bajo nivel, pues las otras características de mayor nivel, como trayectorias, serán más sensibles a fenómenos de iluminación, oclusiones... [1] siendo posible su inclusión posterior sobre los sistemas desarrollados.

2.1 Método de aprendizaje

El método de aprendizaje aplicado a la detección de anomalías es el modo que tiene el sistema de, a partir de unos datos que se le presentan, ser capaz de diferenciar cuáles de ellos pertenecen a un comportamiento normal y cuáles son anómalos. Se trata por tanto de un problema de Inteligencia Artificial (IA) de toma automática de decisiones binarias (1/0, normal/anómalo), para lo que será necesario que el sistema sea entrenado.

Atendiendo a la existencia o no de conocimiento *a priori* y de la involucración humana que se desee o se permita en este proceso se puede distinguir entre aprendizaje supervisado y aprendizaje no supervisado (se denominará semisupervisado cuando presente características de ambos).

Utilizando como base los documentos [3, 2, 1] y los conocimientos adquiridos a lo largo del grado en tratamiento de señal e IA:

2.1.1 Supervisado

Es la aproximación más sencilla al problema planteado, pues se basa en datos etiquetados manualmente por el ser humano. El sistema aprende a partir de unos datos de entrenamiento, *training*, de los que previamente se conoce su clasificación (etiqueta) correcta, que se pasan como entrada. Una vez realizada la fase de entrenamiento del detector el sistema podrá clasificar los nuevos datos de entrada, *test*, en cada una de las clases que se hayan definido.

Esta aproximación permite conseguir tasas de acierto elevadas con una dificultad baja, pues simplemente hay que comparar las nuevas muestras de entrada con las etiquetas existentes y asignarlas a una de las categorías. Por el contrario, la dificultad de su perfecta implementación es muy elevada incluso para secuencias con pocos eventos anómalos, pues es necesario tener muestras de cada uno de los posibles comportamientos presentes en la escena, y además las personas pueden realizar una misma acción de multitud de formas distintas.

El algoritmo más utilizado de aprendizaje supervisado es el algoritmo de aprendizaje por arboles de decisión, *decision tree learning*, y sus diferentes variantes [1].

Por todo ello, este método de aprendizaje únicamente será válido si los eventos anómalos están perfectamente definidos y hay suficientes datos de entrenamiento para definir todos los posibles eventos que puedan ocurrir, que por lo general no será el caso, lo que no quiere decir que no tenga aplicaciones. Un ejemplo válido sería un reconocedor de gestos en el que estos están limitados, como el presentado en [19].

2.1.2 No supervisado

Este es el caso opuesto al aprendizaje supervisado, pues supone el aprendizaje automático del sistema mediante la entrada de datos observados sin etiquetar, basándose en sus propiedades estadísticas. De esta manera, en cuanto a detección de anomalías, clasificará como tales a aquellos eventos que estadísticamente tengan poca representación que aparecerán como agrupamientos o clústeres aislados, es decir, los que ocurran menos frecuentemente.

La gran ventaja de este tipo de aprendizaje frente al supervisado es que se podrán modelar automáticamente un número muy alto de posibles eventos y permitirá abordar por tanto el análisis de escenas reales de comportamiento humano en entornos de videovigilancia sin la necesidad de tener muestras de todos y cada uno de los eventos que puedan darse..

A la vista de esto, y dado que se dispone de bases de datos en las que se tienen secuencias de entrenamiento con etiqueta de normalidad pero se carece de muestras de anomalías debidamente etiquetadas, se utilizará en este trabajo un método de aprendizaje semi-supervisado desde el punto de vista de la disponibilidad de secuencias de entrenamiento [20].

2.2 Modelado

Una vez se tienen calculadas las distintas características que se extraen para interpretar el comportamiento de la escena y estudiado el método de aprendizaje, el siguiente paso es crear modelos de normalidad que permitan clasificar los eventos. Para ello lo que se propone es extraer, a partir de dichas características, una serie de patrones de movimiento, que se definen en [2] como regiones de la escena con una gran similitud en sus características y suficientemente distintas a las demás regiones y que describen tanto la segmentación espacial como la tendencia del movimiento en un periodo de tiempo.

Una vez extraídos los patrones de movimiento, históricamente se han propuesto en la literatura métodos para clasificarlos basados en reglas y métodos estadísticos basados en probabilidad [3], siendo los segundos los que se emplearán pues permiten construir modelos de actividad menos limitados que el mero uso de reglas predefinidas. Dentro de estos, se puede distinguir entre métodos que aprenden previamente un modelo de normalidad, anomalía o ambos simultáneamente para decidir después, conocidos como *offline*, y modelos que automáticamente van aprendiendo los patrones presentes en las secuencias [3], *online*. Por simplicidad y por compatibilidad con las secuencias que se van a emplear, se recurre en este trabajo al modelado *offline*.

De acuerdo con el principio de segmentar, clasificar o *clusterizar* los patrones, se han propuesto diferentes métodos que pueden agruparse en las tres categorías que se describen a continuación.

2.2.1 Segmentación basada en modelos de flujo de campo

En estos modelos se trata a la multitud en movimiento como un flujo que depende del tiempo. Se divide la escena en regiones que sean dinámicamente distintas y se trata de agrupar los patrones de movimiento de ellas con propuestas entre las que destacan las basadas en mecánica lagrangiana (una reformulación de la mecánica clásica), empleando para ello algoritmos basados en bordes, grafos o vertientes (*watershed*) [2].

Se ha demostrado que estos modelos ofrecen gran rendimiento en escenas de alta densidad pero, al ser características basadas en píxeles como lo es el flujo óptico, no se podrán manejar eventos con solapamiento de patrones. Además, no funcionarán cuando la densidad de la escena sea baja, y eso es algo que no se puede suponer en este trabajo pues se cumplirá para algunas pero no para todas las secuencias analizadas. Por ello, y dada la dificultad que entrañan y que no se emplearán por el tipo de características elegidas, no se profundizará más allá en ellos.

2.2.2 *Clustering* basado en similitud

Se aborda la segmentación de patrones de movimiento como un problema de agrupamiento o *clustering* en tres pasos [2]:

1- Extracción de características:
2- Agrupamiento de características según criterios de similitud;
3- Estimación de las regiones de normalidad/anomalía a partir de las agrupaciones.

Documentos como [21] emplean y comparan el funcionamiento de dichos modelos, entre los que destaca el algoritmo *K-Means*, cuya implementación para procesamiento de imagen se explican en detalle en [16] y puede verse en el Anexo 0. Además, en [22] se estudian distintas medidas de similitud aplicables a estos modelos de *Clustering* más comunes, analizando sus ventajas e inconvenientes, los detalles están disponibles en el Anexo 0**.**

Adicionalmente, en varias ocasiones se hace mención en la literatura a los Modelos Ocultos de Markov o HMMs (*Hidden Markov Models*) junto con estos métodos de *Clustering* en cuanto a la segmentación de patrones movimiento para la detección de anomalías [1, 3] .

2.2.3 *Clustering* basado en modelos probabilísticos

El agrupamiento (*clustering*) basado en modelos probabilísticos permite hacer un salto cualitativo importante a la hora de analizar el comportamiento de una escena, pues se puede hacer un análisis a largo plazo que no se podía hacer con los métodos vistos hasta el momento y facilita la comprensión del movimiento de la multitud [2]. Estos modelos nacen de la adopción de modelos bayesianos de probabilidad para el agrupamiento de las características extraídas, y su gran ventaja es que generan una representación mucho más compacta que el agrupamiento directo por similitud.

.

Entre las múltiples propuestas de modelos probabilísticos, los más empleados en la literatura para la detección de anomalías son el Modelo de Mezcla de Gaussianas o GMMs (*Gaussian Mixture Models*), RFTs (*Random Field Topics*) y LDAs (*Latent Dirichlet Allocation*) [2, 1], y de ellas, por hacer sido estudiadas en el grado y tener un grado alto de asimilación de las mismas y para limitar su extensión, planteará este trabajo únicamente en el primero de ellos.

GMM defiende que cualquier distribución probabilística se puede aproximar y modelar mediante una serie, más o menos amplia, de distribuciones normales o gaussianas y, lo que se propone por tanto, es modelar los eventos de la escena de normalidad y/o anomalía en función de este tipo de distribuciones. Es un método ampliamente utilizado y estudiado y, aunque en [2] se entiende el uso de GMMs sobre el flujo óptico, este no está limitado únicamente a estas características, sino que se pueden adaptar a cualquier vector de características que se elija para representar la normalidad, como serán las características espacio-temporales locales basadas en el gradiente en este caso.

Como conclusión, se ha presentado hasta ahora un resumen de los conceptos, métodos y aplicaciones de la literatura, destacando cuáles se han considerado más relevantes para este trabajo y se van a utilizar, por lo que el siguiente paso será la implementación particular realizada de lo visto hasta ahora.

3 Diseño del sistema

En la figura 3-1 se muestra la estructura de bloques general del detector de anomalías en vídeo que se propone en este trabajo, basado en el diagrama de flujo propuesto por [1] y adaptado a una estructura funcional de bloques. Como se puede ver, tiene dos entradas: la primera para las secuencias de entrenamiento, que se utilizarán para crear un modelo de comportamiento, y la segunda para las secuencias de test a partir de las cuales se genera una salida que debe ser una decisión sobre la normalidad o anomalía de la misma, compartiendo ambas un mismo bloque extractor de características. Además, existe un elemento previo al detector y que funciona como entrada al sistema, consistente en una serie de consideraciones previas y necesarias para su correcta implementación.

A continuación, se analizarán en detalle todos los bloques y se realizará su implementación individual para este trabajo.

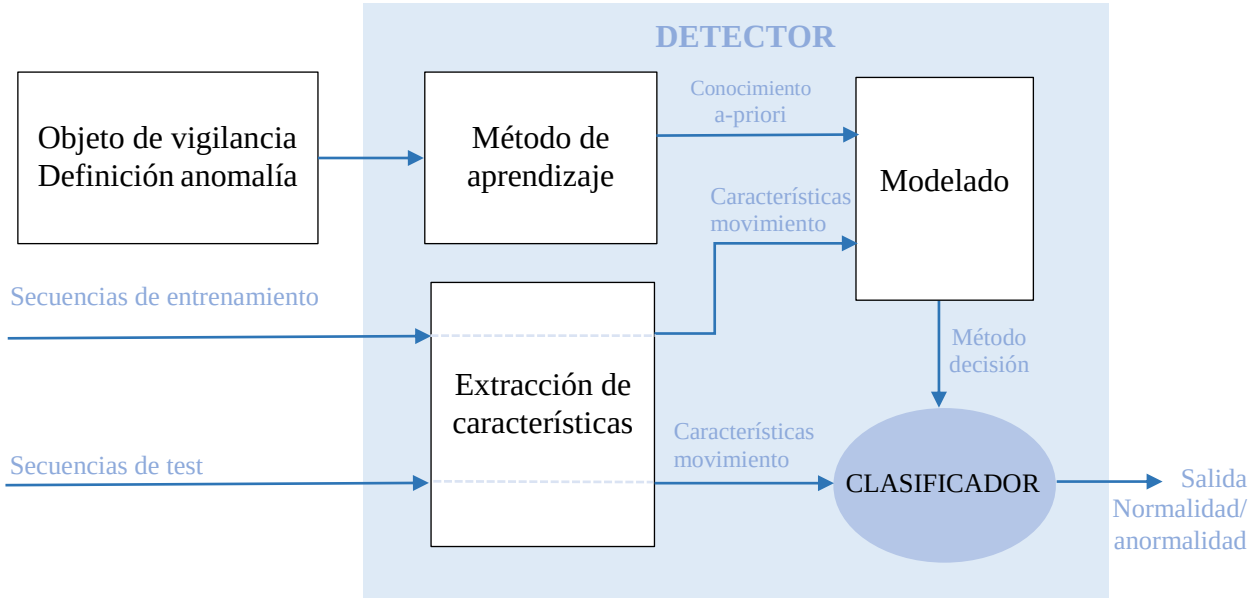


Figura 3-1: Estructura general del detector

3.1 Objeto de estudio y anomalía

El objeto de este trabajo es la clasificación automática de una secuencia de vídeo según responda en el tiempo a un comportamiento normal o anómalo, como se muestra en la Figura 3-1, donde las personas marcadas en rojo están corriendo. Este problema, que puede parecer trivial dada su descripción, es complejo desde el inicio pues una situación puede considerarse anómala o no dependiendo del contexto (en otra situación puede ser normal que los individuos corran), por tanto se debe primero intentar contextualizar las escenas que se estudiarán y definir qué es una anomalía en las mismas, extrayendo unas primeras conclusiones sobre cómo será el comportamiento de las escenas para poder modelarlas más eficaz y eficientemente y ser capaz de detectar dónde y cuándo ocurren las anomalías reduciendo al máximo los errores.

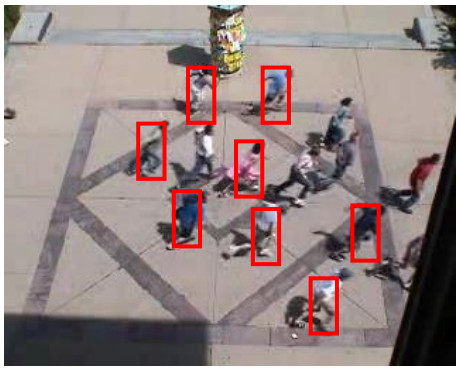


Figura 3-2: Ejemplo de anomalías en el dataset público UMN

Como se ha visto en el estado del arte, las escenas para las que se va a implementar el detector serán escenas de multitudes con una densidad variable desde uno o ningún individuo hasta varias decenas, en las que las personas serán los agentes (elementos en movimiento).
El objetivo final es el de detección de anomalías en entornos de videovigilancia de escaleras mecánicas, en los que el nivel de densidad será muy variable (desde nulo todo el tiempo en el que no haya nadie en las escaleras hasta una densidad muy alta en "horas puntas") y se puede suponer una iluminación uniforme y controlada por tratarse de escenas de interiores. Sin embargo, como también se harán pruebas con bases de datos públicas de otro tipo de escenas (de exteriores y con vistas muy distintas), no se hará ahora ninguna suposición de este tipo.

3.2 Método de aprendizaje

Como se ha comentado en el estado del arte, el aprendizaje o entrenamiento del sistema se lleva a cabo de una manera semi-supervisada, generando un modelo a partir de secuencias de entrenamiento etiquetadas como normalidad.

Durante la etapa de entrenamiento, se seleccionan una o varias secuencias en las que se conoce *a priori* que no existe ninguna anomalía, pero se desconocen (no están etiquetados) los eventos que en ellas ocurren. Se extraen las características descritas anteriormente y se crea a partir de ellas un modelo de normalidad que sirva como base para detectar las anomalías. Por ello, al ser tan diferentes los tipos de escena que se analizarán, se realizará un entrenamiento para cada una de ellas, es decir, para detección de anomalías en secuencias de escaleras mecánicas con vista cenital se entrenará con secuencias de normalidad de esta escena, para una vista superior y más alejada lo mismo, y así sucesivamente.

La consecuencia de que los eventos sean diferentes en cada escena y de este proceso de aprendizaje es que el sistema entrenado no será genérico, sino único para cada tipo de secuencia, y se necesitarán secuencias de entrenamiento para cada escena que se quiera modelar. Sin embargo, se considera acertado para este trabajo pues se disponen de dichas secuencias y se aumentará de esta manera sensiblemente el rendimiento del sistema.

Se procede ahora a la implementación particular del detector propuesto, formado por tres bloques principales: extracción de características, método de aprendizaje, algoritmo de modelo y el clasificador.

3.3 Extracción de características

La clave para que el modelado sea exitoso es que las características elegidas sean altamente descriptivas y discriminativas para una representación de actividad mínima y compacta. Como se ha visto en [3] las características locales espacio-temporales tienen un gran poder descriptivo, y por ello para este trabajo se van a emplear características basadas en los gradientes de los valores de intensidad. Por tanto, se está ante un bloque extractor a nivel de pixel de características locales espacio-temporales basadas en los gradientes de la secuencia.

El bloque de extracción de características de la secuencia entrada tiene la estructura interna mostrada en la Figura 3-3, que se muestra a continuación:

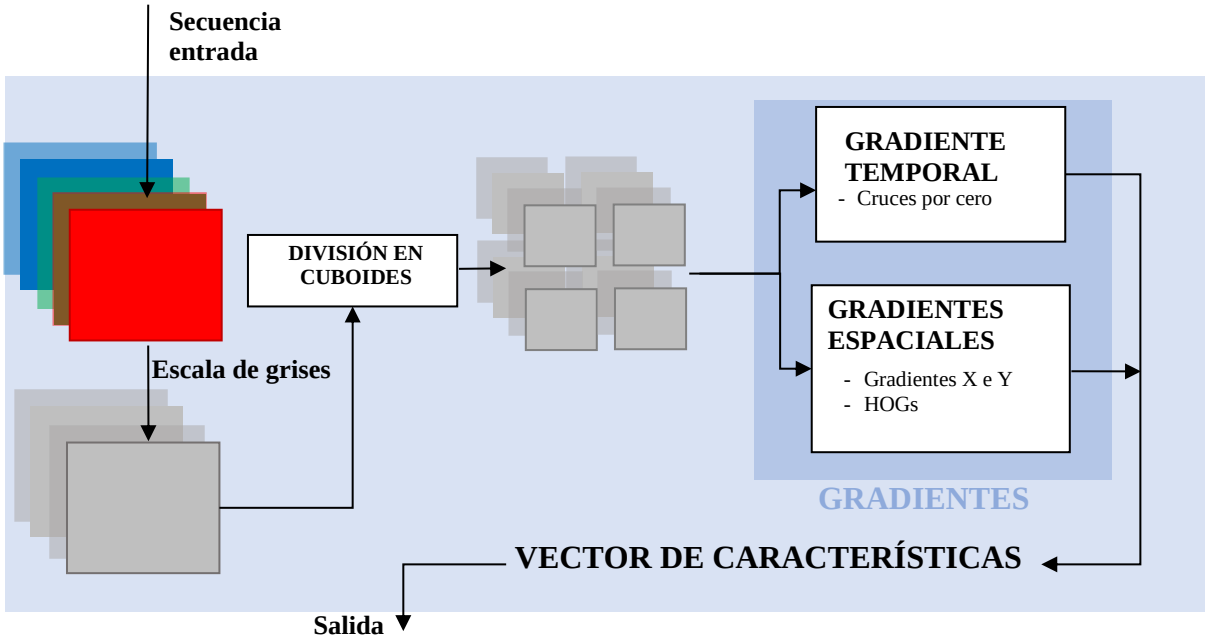


Figura 3-3: Estructura bloque de extracción

Se describen ahora cada uno de los puntos de esta estructura:

- Entrada: La entrada al bloque extractor de características es el vídeo como secuencia temporal de imagenes. Puesto que el formato del mismo puede ser a color, y no se necesita esta información, se lleva a cabo una conversión a escala de grises que asegure la compatibilidad del sistema para cualquier secuencia y se reduzca la complejidad y carga computacional del mismo.
- División en cuboides: a la hora de analizar la secuencia, y como se hace en todos los trabajos relacionados, se ha de dividir esta en diferentes subsecuencias espacio-temporales llamadas cuboides para extraer las características locales. La elección de tamaño de estos cuboides será crucial para el cálculo de los gradientes y por tanto para el funcionamiento del detector y se analizará en detalle en apartados posteriores. El objetivo de estos cuboides es dividir la escena en conjuntos que engloben eventos locales y permitan modelar de la mejor forma posible el comportamiento de los mismos.
- Gradientes: Este es el bloque en el que realmente se extraen las características para modelar los eventos. Como se ha explicado, estas son características espacio-temporales basadas en los gradientes.

La obtención de gradientes obedece a la siguiente fórmula:

$$\nabla \mathrm{I} = \left[I_x, I_y, I_t\right]^T = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}\right] \quad (1)$$

Donde $I_{x;y;t}$ es la derivada en cada dimensión de los píxeles del cuboide. En la práctica, en la literatura y en este trabajo, se realizan distintas implementaciones y cálculos para la obtención de los mismos.

3.3.1 Gradiente temporal

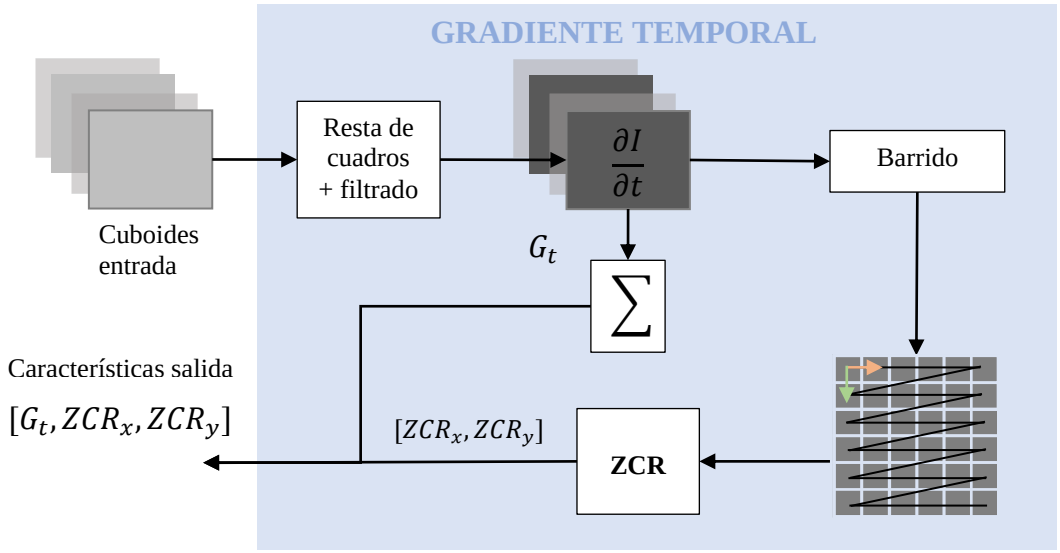


Figura 3-4: Bloque extractor Gradiente Temporal

En la figura superior se muestra la estructura interna del bloque extractor del gradiente temporal en el sistema propuesto que, como se puede observar, está a su vez dividido en bloques con una funcionalidad más limitada.

El gradiente temporal se define como la derivada respecto al tiempo de los pixeles contiguos en el eje temporal. Para implementarlo, se elabora una aproximación por medio de la resta de píxeles consecutivos en esta dimensión, como se muestra en la fórmula (2).

$$\frac{\partial I}{\partial t} = I_t - I_{t+1} \quad (2)$$

En este trabajo se no se realizará un filtrado de los valores de gradiente temporal de intensidad, pero puesto que el vídeo es una señal muy susceptible a presentar ruido, como trabajo futuro será muy interesante estudiar tanto este filtrado como el de los gradientes espaciales, para ver cómo puede mejorarse la detección y robustez del sistema gracias a ello.

De esta forma se obtiene un cuboide diferencia o cuboide de gradiente temporal, con valores en cada píxel. Para extraer un valor de gradiente temporal por cuboide, lo que se propone y se implementa es el sumatorio de todos los píxeles diferencia. Siendo X e Y el tamaño del cuboide:

$$G_t = \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} \frac{\partial I}{\partial t}\Big|_{i,j} \quad (3)$$

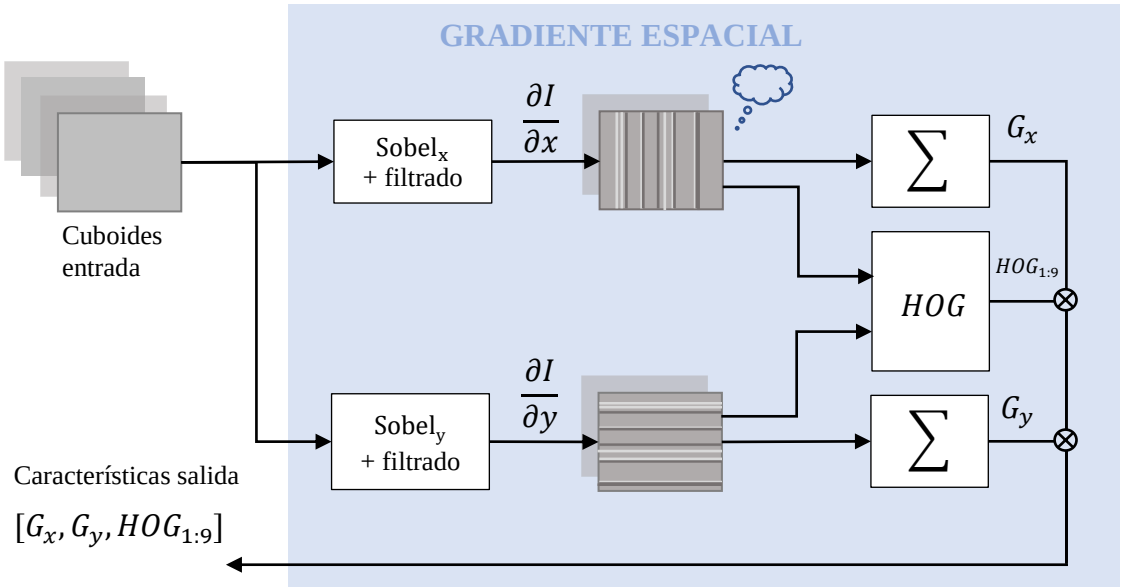
Dado que se piensa, y se demostrará más adelante, que esta interpretación del gradiente temporal en ocasiones puede aportar información muy pobre para la detección de anomalías (aunque como se verá en la fase de pruebas, generalmente será muy descriptivo), se propone en este trabajo utilizar además una tasa de cruces por cero o *zero crossing rate*, (ZCR) sobre el cuboide de gradiente temporal. Esta tasa, muy utilizada en tratamiento de audio, indica el número de veces que una señal atraviesa el valor de cero, por lo que aproxima una idea de frecuencia que podría traducirse semánticamente a "velocidad de cambio" en la región analizada. Se produce un cruce por cero cuando o bien dos muestras consecutivas muestran distinto signo, o bien una muestra equivale a cero. Así:

$$ZCR = \sum_{m=1}^{N} |sgn(x[m]) - sgn(x[m-1])| \quad (4)$$

siendo N el número de muestras de la señal $x[m]$. Para adaptar esta ratio a los cuboides de gradiente temporal calculados, se procede como se muestra en Figura 3-4.

1- Se inicializan a cero las tasas ZCR_x y ZCR_y.
2- Se selecciona un píxel del cuboide y sus adyacentes derecho e inferior.
3- Se compara el primer píxel y se suma 1 a ZCR_x y ZCR_y si el signo de los píxeles es distinto, respectivamente.
4- Se repiten los pasos 2 y 3 para todos los píxeles del cuboide, haciendo un barrido para recorrerlo, hasta llegar al último.

3.3.2 Gradientes espaciales



Los cuboides resultantes de la aplicación de los gradientes en *x* y en *y* mostrarán, respectivamente, los bordes verticales y horizontales del cuadro.

Figura 3-5: Bloque extractor Gradientes Espaciales

La Figura 3-5 muestra el proceso de extracción de características del gradiente espacial llevado a cabo en la implementación del detector de anomalías. Juntando todas las características extraídas, que son G_x, G_y y , $HOG_{1:9}$, se puede ver que el vector descriptor del gradiente espacial tiene un tamaño 11 (2 de los gradientes y 9 del HOG), que se estudia en la parte de análisis de resultados.

Los gradientes espaciales se definen como la derivada respecto a x e y de los píxeles del cuboide. Para su implementación se ha optado por el uso del operador Sobel, muy empleado en toda la literatura de tratamiento de imagen, como se propone en [23].

$$\frac{\partial I}{\partial x} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \tag{5}$$

$$\frac{\partial I}{\partial y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \tag{6}$$

Así, al igual que ocurría con el gradiente temporal, se obtiene un cuboide de gradiente de la aplicación de la máscara de Sobel. Por tanto, en este caso se generarán dos cuboides de gradiente en *x* y en *y*, y se calculará el valor de gradiente final mediante el sumatorio de los cuboides de gradiente:

$$G_x = \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} \frac{\partial I}{\partial x}\Big|_{i,j} \tag{7}$$

$$G_y = \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} \frac{\partial I}{\partial y}\Big|_{i,j} \tag{8}$$

Con X e Y definidos como el tamaño del cuboide.

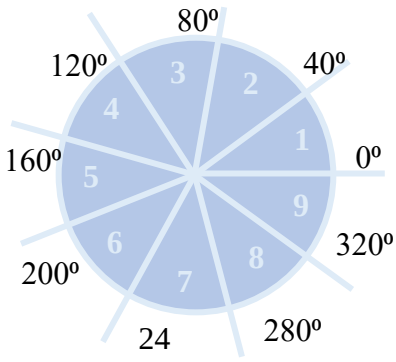
Puesto que el gradiente espacial es una aproximación a la derivada de la intensidad de la imagen en diferentes direcciones, que marca los bordes o contornos presentes en la imagen, este cálculo puede arrojar una primera idea también en cuanto a las orientaciones existentes. Como se considera que estos gradientes calculados sobre las divisiones de la secuencia en cuboides tendrán información muy relevante para la detección de anomalías en cuanto a magnitud y orientación del movimiento presente, se obtiene a partir de ellos el Histograma de Gradientes Orientados o HOG (*Histogram of Oriented Gradients*) de 9 intervalos para intentar extraer una información más fina de los mismos, pues se demuestra en [24] el buen funcionamiento de este descriptor para la detección de personas (aunque la implementación permite cambiar manualmente esta cantidad de bines).

De tal forma las características se calculan como se describe a continuación:

1- Se calcula la magnitud y la fase de la orientación de cada píxel mediante un cambio de coordenadas cartesianas a polares:

$$\left(\frac{\partial I}{\partial y}\right)_{x,y} \xrightarrow{polar} \left(\frac{\partial I}{\partial y}\right)_{r,\theta} \quad (9)$$

2- Se compara la coordenada θ de cada píxel del cuboide para decidir a cuál de las divisiones de la circunferencia mostrada en Figura 3-6 corresponde.



Orientación	Ángulos
1	[0, 40)
2	[40, 80)
3	[80, 120)
4	[120, 160)
5	[160, 200)
6	[200, 240)
7	[240, 280)
8	[280, 320)
9	[320, 360)

Figura 3-6: Orientaciones Gradiente 9 intervalos

3- Se hace, para cada intervalo, el sumatorio de la magnitud r de los píxeles que contiene. Con ello se busca asignar pesos a las muestras del histograma directamente en función de su módulo para disponer de más información sobre la magnitud del movimiento, además de su orientación.

Finalmente, el vector de características será la agregación de un vector de tamaño 3 para las características temporales y 11 para las espaciales, que juntos conforman el vector de características de tamaño 14 que se empleará a lo largo del análisis de los resultados.

$$[G_t, ZCR_x, ZCR_{y,}\, G_x, G_y, HOG_{1:9}]$$

3.1 Modelado

En este apartado, se van a estudiar los distintos modelos propuestos para interpretar las características extraídas de las secuencias y decidir sobre su normalidad o no normalidad. Lo que se propone en este trabajo es realizar una transición desde un modelo de *Clustering* puramente basado en similitud hasta otro, también basado en similitud, pero que permita introducir dispersión, como lo es *Fuzzy K-Means* o *Fuzzy C-Means* (FCM), que permitirá modelar de una manera más realista las escenas. Se intentará con ello ver qué ventajas pueden aportar y qué inconvenientes pueden producir cada uno de los algoritmos en el detector.

3.1.1 K-Means

Para la primera aproximación de modelo de clasificación para la detección de anomalías se ha elegido, tanto por conocimiento previo del mismo como por presencia en la literatura estudiada [21], el algoritmo de *Clustering* basado en similitud *K-Means*. El funcionamiento formal del mismo y su formulación se explican en el Anexo 0.

Para entender cómo será el entrenamiento, se muestra un ejemplo sencillo de *Clustering* en el que, para su representación se ha reducido el número de características empleadas a 2 (que podrían ser, por ejemplo, los gradientes *Gx* y *Gy*) y el número de *Clusters* también a 2. Al estar usando un vector de dos características, se genera un espacio bidimensional en el que cada muestra se corresponde con un cuboide. Para comenzar se puebla este espacio con las características de la secuencia de entrenamiento, como se muestra en Figura 3-7.

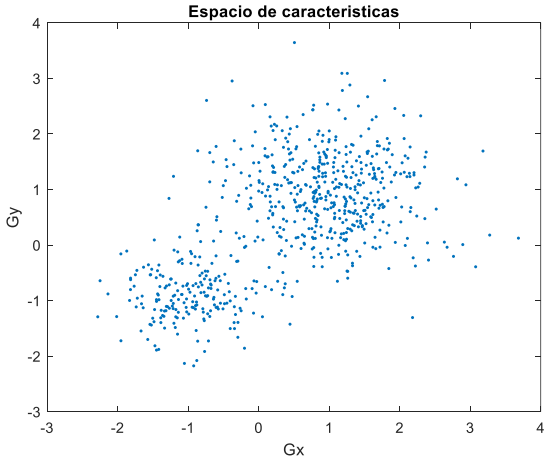


Figura 3-7: Ejemplo espacio vectorial Gx y Gy

Se inicializan los centroides dentro del espacio. Como en este caso sólo hay dos *Clusters*, existirán dos únicos centroides, que en cada ejecución del algoritmo se colocarán inicialmente en un lugar del espacio diferente.

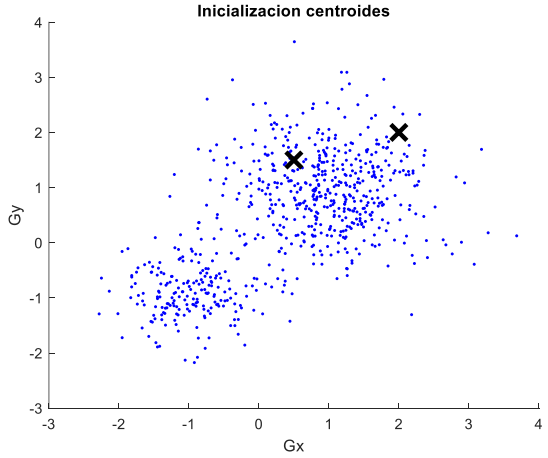


Figura 3-8: Ejemplo inicialización centroides

Entre las opciones disponibles para esta inicialización, se ha optado en el trabajo por la utilización del método heurístico del algoritmo *K-Means++*, que mejora el tiempo de ejecución del algoritmo original (Lloyd) inicializando los centroides lo más alejados posible, dando lugar a los mismos resultados [25]. Este centroide se irá actualizando recursivamente buscando minimizar la función de distancia elegida, que en este caso será la euclídea (aunque sería interesante probar otras distancias que tuvieran en cuenta la dispersión de las muestras, ver en Anexo D) hasta alcanzar el criterio de parada.

Como medida adicional para asegurar la convergencia del algoritmo a un mínimo global y no local, se repite el algoritmo desde la inicialización en varias ocasiones, cinco concretamente, y se selecciona aquella iteración que minimice más la función de distancia. Se muestra en Figura 3-9 cómo quedaría dividido el espacio.

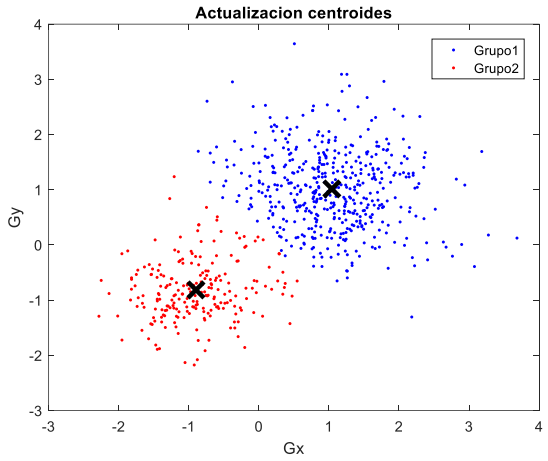


Figura 3-9: Ejemplo actualización centroides

Así, una vez optimizados los centroides para el espacio de características de normalidad de la escena, el entrenamiento del sistema estará completo y el bloque clasificador podrá tomar la decisión sobre anomalía o no en función de cómo se vayan situando en el espacio las nuevas muestras de entrada sin etiquetar.

3.1.2 Fuzzy C-Means

Fuzzy C-Means, o FCM, es una evolución del algoritmo *K-Means* que se acaba de describir en la que la pertenencia a cada una de las clases no es binaria, sino que se tiene un grado de pertenencia a cada *Cluster*. Es, por tanto, un algoritmo de *Soft Clustering* frente al *Hard Clustering* planteado hasta el momento.

El principio de clasificación e iteración del algoritmo es el mismo, pues la función de minimización únicamente añade un término de matriz de partición $(U)^m$, que indica el grado de pertenencia de la muestra a cada uno de los centroides existentes con valores entre 0 y 1, debiendo sumar 1 en su totalidad. Su exponente m que indica el grado de difusión del algoritmo desde 1 (difusión nula, *Hard Clustering* como en *K-Means*) hasta el infinito, aunque nunca se toman valores mayores de 30 (*Soft Clustering*), siendo 2 el factor elegido por ser el utilizado en la mayoría de las aplicaciones [26].

Por ser muy similar, aquí se tratará de intentar mostrar las diferencias frente a *K-Means* más que estudiar su funcionamiento paso por paso (se puede consultar la explicación formal y la formulación en Anexo C). De esta manera, y como se demuestra en [26], esta nueva implementación clasifica mejor los casos reales, pues no asigna las muestras a un único *Cluster*, se adapta mejor a las distintas formas que estos puedan presentar y es menos sensible al ruido y los *outliers*. Como contras, presentará un tiempo de ejecución mucho más alto y habrá que tener consideraciones como que el número mínimo de *Clusters* será 2, mientras que antes era 1, y se deberá diseñar un bloque de decisión diferente.

Por todo ello, se considera una opción muy válida para este trabajo y se implementan un módulo de entrenamiento y otro de decisión, que se describirá después, mediante este método.

3.2 Clasificador

El módulo clasificador es el encargado de decidir, a partir del modelo de normalidad aprendido, cuándo una muestra de entrada será considerada anómala o no. Esto se logrará mediante una comparación directa con el modelo aprendido: cuando la muestra se corresponda con alguno de los grupos generados en el modelo (*clusters* para *K-Means* y FCM) será una muestra de normalidad, mientras que cuando ésta no pueda asociarse a ninguno será considerada anómala.

Cómo medir esta correspondencia no es un asunto trivial y es el objetivo de este apartado, pues será crítico para el correcto funcionamiento del sistema y dependerá del algoritmo utilizado para generar el modelo. Así:

a) *K-Means*

La decisión se tomará en función de la distancia (euclídea, aunque existen otras como se ha comentado) que separe la muestra con los centroides de los *clusters* del modelo, cuanto mayor sea esta distancia más anómala se considerará la muestra, y viceversa.

Formalmente, el cálculo particular realizado es el que se muestra a continuación acompañado de las líneas de código que lo implementan.

Se obtiene distancia de las muestras del modelo con los centroides de los *clusters* a los que están asignadas.

```
min_distancias_train = min(distancias_train, [], 2);
```

Donde 'distancias_train' son las distancias de cada muestra de entrenamiento a todos los clústers y los argumentos de la función indican la dimensión en que se calcula este mínimo.

Se calcula la distancia media de las muestras de entrenamiento a sus respectivos *clusters*. Se ha escogido la media frente a otras opciones que podrían parecer más lógicas, como el máximo, para intentar mitigar en cierta medida el efecto de los valores ruidosos y extremos.

```
mean_distancias_train = mean(min_distancias_train);
```

Se obtiene el umbral entre anomalía y normalidad a partir de la distancia media, añadiéndole esta distancia por un factor 'porcentaje_distancia_anomalia', que se podrá variar para ajustar el umbral.

```
umbral = mean_distancias_train +
(porcentaje_distancia_anomalia/100).*mean_distancias_train;
```

Ahora, podrán considerarse anómalas aquellas muestras de entrada cuya distancia al centroide más cercano supere el umbral establecido.

Este procedimiento para la toma de decisión tiene la gran ventaja de que es sencillo y muy intuitivo, por lo que se ha usado como primera aproximación para el detector a expensas de analizar los resultados que ofrezca en el apartado de análisis. No obstante, ya se puede saber que su gran limitación va a ser la presencia de ruido y de valores extremos en el entrenamiento de normalidad.

b) *Fuzzy C-Means*

En caso de utilizar este segundo algoritmo para modelar la normalidad en la escena el de umbral para decisión visto hasta ahora no será válido, pues el resultado de este no son unas distancias sino unos grados de pertenencia a cada *cluster* y esto hace el proceso algo menos intuitivo.

Idealmente, una muestra será anómala cuando no pueda asociarse a ninguno de los *clusters* generados en el entrenamiento, pero en la implementación este cálculo no será tan sencillo pues el algoritmo siempre asigna a cada muestra un valor de pertenencia a todos *clusters* y estas suman 1. De esta manera, lo que se ha concluido es que la muestra no pertenecerá a ninguno de los grupos cuando pertenezca a todos por igual (por ejemplo, para 2 *clusters*, pertenencias 0.5 y 0.5), algo que en el espacio de características ocurrirá o bien cuando la muestra se aleje mucho de todos los centroides, o bien cuando la misma se sitúe en el punto medio entre ellos, es decir, cuando las muestras se sitúen en el hiperplano que está a la misma distancia de todos los centroides.

Visto teóricamente cómo se hará esta decisión, se procede a explicar los cálculos específicos realizados para implementarlo partiendo de la matriz 'U_decision', matriz de pertenencia de las muestras de entrada a los *clusters* generados en el entrenamiento.

Para calcular que los valores son los mismos para todos los grupos se calcula la derivada de primer orden, que indicará si la curva de pertenencias es plana (igual pertenencia) o presenta máximos y mínimos (mayor pertenencia a alguno de los *clusters*).

```
dU_decision = diff(distancias_decision);
```

Se calcula el grado de similitud como el sumatorio del valor absoluto de la derivada que se acaba de obtener, con el objetivo de tener un resultado que sea cero cuando las pertenencias sean exactamente iguales y vaya creciendo conforme se hacen más dispares.

```
similitud = abs(sum(dU_decision,1));
```

Con este valor de similitud ya se procede a comparar con el umbral, que en el mejor de los casos será 0 pero se permite que varíe entre 0 y 1 obtener curvas de rendimiento y poder ajustarlo al mejor funcionamiento del detector.

De esta manera, queda ya expuesta la totalidad de la implementación individual de los bloques del detector utilizados en este trabajo y se ha de pasar a la fase de pruebas y análisis de resultados.

4 Pruebas y análisis de resultados

Como en todo sistema, una vez realizada la implementación se han de realizar una serie de pruebas para comprobar que su funcionamiento es correcto y acorde a los requisitos iniciales. Además, dado que en este diseño existen multitud de parámetros que se pueden variar, desde el tamaño de los cuboides hasta el umbral de decisión pasando por las posibles características a utilizar, el número de grupos en el entrenamiento…, esta fase de pruebas servirá para analizar el rendimiento del sistema para cada conjunto de parámetros, con sus ventajas en inconvenientes, y así poder optimizar el sistema con el criterio que se decida. En este caso el criterio será el de tasa de detección por tratarse de un sistema sencillo con poca algoritmia y datos, pero en un sistema con un volumen de cálculo más grande habrá que tener en cuenta otros criterios como el tiempo de ejecución y buscar un compromiso entre ambos.

En este apartado, por tanto, se describirá el método de evaluación y el protocolo de pruebas seguidos para este análisis y se procederá, secuencia a secuencia, a estudiar el rendimiento del sistema con distintas especificaciones.

4.1 Ground-Truth

Para poder realizar medidas objetivas de aciertos y errores del sistema se ha de tener una referencia, *Ground-Truth*, de las anomalías reales presentes en cada una de las escenas analizadas. Con este fin, se ha creado una herramienta que permite manualmente ir etiquetando cuadro a cuadro el contenido de un clip de vídeo, marcando con el cursor del ratón aquellas zonas de los cuadros donde existan anomalías.

De este modo se genera, para cada secuencia de test, una máscara de referencia en la que las anomalías aparecen como 1s y la normalidad como 0s, lo que permitirá comparar automática y objetivamente los resultados obtenidos por el sistema con la realidad de la escena. Se hacen un total de 7 secuencias de referencia cuadro a cuadro, 3 de la base de datos pública UMN y 4 de secuencias de escaleras mecánicas con anomalías.

Este proceso, aunque sea lento, permite obtener una referencia tan precisa como se desee y cuidadoso se sea a la hora de etiquetar las anomalías. Este no será un problema para este trabajo dado que el análisis de rendimiento se hace a nivel de imagen, es decir, se consideran cuadros anómalos o normales sin localizar las anomalías dentro de los cuadros, por lo que el etiquetado podrá realizarse de manera más tosca. En el momento en que se encuentre un cuboide anómalo en un cuadro de la escena el cuadro quedará clasificado como anómalo.

4.2 Medidas de evaluación

Como se acaba de comentar, la evaluación de las pruebas y su rendimiento será realizada a nivel de imagen. Esto hará el proceso más sencillo, teniendo en cuenta que tendrá limitaciones pues no se podrá asegurar que las anomalías encontradas correspondan realmente con las presentes.

Para evaluar cuán bueno es un resultado se van a calcular curvas de rendimiento ROC (*Receiver Operating Characteristic*) que relacionan la FPR (*False Positive Rate*) y la TPR (*True Positive Rate*) en los ejes X e Y, respectivamente.

Estas curvas se obtienen a partir de las tasas de acierto y error producidas en las distintas ejecuciones del detector al variar el umbral de decisión (distancia euclídea para KM, grado de pertenencia para FCM). Estas tasas se definen en la Tabla 4.1

Tabla 4.1 Tasas de Acierto y Error

		DETECCIÓN		
		Normalidad	Anomalía	
GROUND TRUTH	Normalidad	TN	FP	$FPR = \frac{\sum FP}{\sum FP + \sum TN}$ (10)
	Anomalía	FN	TP	$FPR = \frac{\sum TP}{\sum TP + \sum FN}$ (11)

Relacionando ambos valores ya se puede calcular, para cada una de las secuencias, para cada conjunto de parámetros fijado, una curva de rendimiento ROC. Estas curvas generalmente tendrán un aspecto como el que se muestra en la Figura 4-2.

Gracias a este tipo de curvas se puede analizar el rendimiento del sistema de forma muy sencilla, pues este rendimiento será tanto mejor cuanta más área quede bajo la curva. Esta métrica es la denominada como *Area Under Curve*, AUC, que variará entre 0 y 1 siendo 0.5 correspondiente a una detección aleatoria de normalidad/anomalía en cada cuadro (representada como la diagonal de la figura en cada ROC adjunta a lo largo de todo el documento). El área descrita se calcula mediante aproximaciones trapezoidales a la curva como se define en (12).

$$AUC = \frac{1}{2}(C1 + C2)(T2 - T1), \tag{12}$$

siendo $(T2 - T1)$ el interespaciado lineal (se emplea 1) y $C1, C2$ los valores de la curva en dichos puntos.

Con todo ello, a partir de las curvas ROC y su AUC se elaborarán otras gráficas que permitan ver la evolución del rendimiento, AUC, en el eje Y en función de los distintos parámetros que se estén analizando en el eje X. Este tipo de curvas se emplean mucho a lo largo del análisis mostrando la evolución del rendimiento con el parámetro del tamaño espacial de cuboide (Anexo A).

4.3 Protocolo de pruebas

Una vez conocidas las medidas de evaluación empleadas, se establece el protocolo a seguir para las pruebas realizadas y el análisis, de manera ordenada, de características y parámetros que optimicen el funcionamiento del detector. Para cada una de las secuencias de test, para cada modelo propuesto, lo primero que se intentará es analizar las características utilizadas y así encontrar un vector de características que sea descriptivo para todas las secuencias, modificando el que se obtiene a la salida del bloque extractor si es necesario, ya sea eliminando algunas o combinándolas, siempre que se demuestre su utilidad. Los pasos que se seguirán con este fin se describen a continuación.

Primero se analizará el gradiente temporal y las tasas de cruces por cero que de éste se extraen para ver su rendimiento y si realmente su implementación realizada aporta información útil para la detección de anomalías. Para ello, se entrenará y probará el sistema

alternando las características usadas G_t, ZCR_x y ZCR_y y se tratará de inferir conclusiones sobre ellas barriendo los parámetros del tamaño espacial y temporal del cuboide empleado. Después se estudiará el gradiente espacial y con éste, por ser una característica derivada, la conveniencia o no del uso de los HOGs como característica adicional. Esto es importante pues es una característica que añade al tamaño del *feature vector* los 9 intervalos del HOG y aumentará por tanto sensiblemente la carga computacional y el tiempo de ejecución, por lo que debe quedar bien justificado si se decide usarlo.

Una vez fijadas las características se acometerá la fijación de los parámetros variables del sistema, que son el tamaño de cuboide (espacial y temporal) y el número de grupos o *clusters* empleados para la clasificación.

4.4 Secuencias analizadas

En este trabajo se van a utilizar dos bases de datos de escenas distintas con el objetivo de ver el comportamiento del sistema implementado en situaciones reales. La primera de ellas, UMN, es una base de datos pública y servirá para comparar el rendimiento de esta implementación con otras de la literatura, mientras que la segunda es una base de datos privada de escaleras mecánicas, que es la escena objetivo del trabajo y de la que no se podrán mostrar imágenes por su carácter privado. Para cada una de estas bases de datos, se tomarán varias de las secuencias de vídeo que ofrecen y se estudiarán en detalle. Aunque se va a estudiar un conjunto de secuencias, aquí sólo se va a mostrar el análisis detallado, con todos los resultados parciales, de la primera de ellas para no exceder el límite de extensión del trabajo. Posteriormente, dado que siempre se seguirá el mismo protocolo de pruebas y obtención de resultados, se explicará únicamente la conclusión final para cada secuencia. Se pondrán en el Anexo A las curvas de rendimiento intermedias en caso de ser necesaria su consulta.

4.4.1 UMN

El *dataset* UMN es un conjunto de secuencias de imágenes de resolución 320x240 píxeles, formado por un conjunto de tres escenas en las que un grupo de personas de densidad media (de una a dos decenas) súbitamente empieza a correr en todas direcciones. Se han utilizado los primeros 10 segundos de cada escena para entrenamiento y el resto para test. Este entrenamiento es pequeño pero, dado que las únicas anomalías existentes se sabe que son por la cantidad y/o velocidad del movimiento (cuando las personas corren), se considera suficiente para la escena. Las imágenes están tomadas desde un punto de vista cenital con poca perspectiva y el fondo (suelo) no es de un color (nivel de gris tras la conversión) homogéneo.

Figura 4-1: Secuencias UMN 1, 2 y 3, respectivamente

Escena 1

La escena 1 tiene la estructura mostrada en Figura 4-1. Durante los 10 primeros segundos que dura el entrenamiento los individuos caminan en solitario o pequeños grupos, situación que continua hasta pasados otros 6 segundos, cuando comienzan a correr en todas direcciones y comienza la anomalía. Este fenómeno se produce un total de 3 veces.

a) K-Means

Se comienzan las pruebas con el modelo generado mediante KM, analizando las características empleadas como se ha descrito en el protocolo de pruebas. Se muestran a continuación los resultados obtenidos con las distintas características obtenidas del bloque extractor en función del tamaño espacial de cuboide, por ser el parámetro más crítico. En un inicio, se fijará el número de *clusters* a 1 para tratar de analizar únicamente el impacto de las características, y posteriormente se pasará a fijar este parámetro. Los resultados de este análisis se encuentran en la Tabla 0.1 del Anexo A.

Si la única característica utilizada para el modelado de la escena es el gradiente temporal G_t, los resultados de rendimiento obtenidos se muestran en la Figura 0-1. Se puede ver en ella que, para esta secuencia, la característica de gradiente temporal es altamente descriptiva, especialmente para tamaños de cuboide pequeños. Esto es así porque los individuos presentes en la secuencia ocupan aproximadamente 20x50 píxeles y las únicas anomalías existentes son porque estos individuos corren. De esta manera, los cuboides pequeños que contengan zonas de las personas como piernas, cabeza… y que permitan construir su silueta, presentarán temporalmente unas diferencias mayores cuanta mayor sea la velocidad a la que se muevan y se podrán detectar como anomalía.

Una vez comprobado que el gradiente temporal tiene un buen funcionamiento, va a analizarse el comportamiento de la tasa de cruces por cero que, a priori, no aportará mucho pues se calcula precisamente para los casos en que el gradiente temporal no ofrezca información significativa. Ejecutando el sistema con el vector de características $[ZCR_x.ZCR_y]$ se obtienen los resultados mostrados en la Figura 0-2, de los que inmediatamente se extrae que en efecto esta característica no es nada descriptiva para esta escena en concreto.

Ahora se pasan a analizar las características espaciales, comenzando por los gradientes y continuando por sus características derivadas como se ha hecho para el caso de las temporales. En la Figura 0-3 se muestran los resultados al emplear la característica $[G_x,G_y]$ como entrada para el clasificador.

Se puede ver que existen picos altos de rendimiento muy bueno, llegando en ocasiones a rozar el AUC de 0.9, y que, como en el caso del gradiente temporal, el funcionamiento es mejor para tamaños de cuboide pequeños (aunque también presenta más variabilidad para ellos). Podría ser una característica útil intentando reducir este carácter ruidoso y eligiendo los pocos tamaños de cuboide con buen rendimiento, pero se probará el HOG para ver si es capaz de mejorar estos resultados.

Se realiza la misma prueba con la característica de los 9 intervalos del HOG calculado y se muestran los resultados en la Figura 0-4, en la que se puede ver que claramente esta característica mejora los resultados obtenidos empleando únicamente los gradientes espaciales. Tiene una curva de valor más alto y menos variable en todo el eje X, y su funcionamiento es especialmente bueno como cabría esperar para tamaños de cuboide pequeños. Además, llama la atención el mínimo existente entre los tamaños de 92x92 y 148x148 píxeles, que se deberá a que estos tienen un tamaño demasiado grande como para

encontrar las anomalías de manera local pero a la vez demasiado pequeño como para hacer un análisis más global de las mismas.

Los resultados obtenidos hasta ahora muestran sin lugar a dudas que para esta escena las características más descriptivas son G_t y HOG, por lo que se crea un vector de características que incluya ambas para ver si son capaces de complementarse y mejorar su rendimiento individual. Se ejecuta el detector con este nuevo vector y obtiene a partir de él la Figura 0-5 Como se puede ver, la inclusión del gradiente afecta positivamente al rendimiento del HOG, de esta manera, queda fijado el vector de características y se ajusta el tamaño del eje de análisis entre 4x4 y 92x92 píxeles. Para ajustar, a partir de aquí, el detector, se va a analizar el impacto del número de *clusters* empleados, pues las distancias y la inicialización ya están fijados.

En la Figura 0-6 se puede ver además que el funcionamiento mejora paulatinamente al introducir nuevos *clusters*, especialmente al saltar de 1 a 2, junto con el tiempo de procesamiento. A partir de ahí mejora más lentamente y para un número mayor de 4 las curvas se superponen, por lo que éste es el valor elegido como óptimo por conciliar rendimiento y tiempo de procesamiento.

De todos los tamaños de cuboide cubiertos en este análisis, se seleccionan 12x12 y 64x64 píxeles: el primero por ser el máximo absoluto y tener un entorno muy parecido, lo que indica estabilidad; y el segundo por alcanzar un rendimiento muy similar con un tamaño de cuboide mucho mayor, de forma que el cálculo necesario es significativamente menor y por tanto también su tiempo de proceso.

Se concluye por tanto, para esta secuencia y para el algoritmo KM, que el funcionamiento óptimo del sistema se tiene para el tamaño de 12x12 píxeles y para las características de gradiente temporal y HOG, y se muestra su curva ROC en la Figura 4-2 junto con la del otro tamaño seleccionado, 64x64, y sus respectivos tamaños en las imágenes.

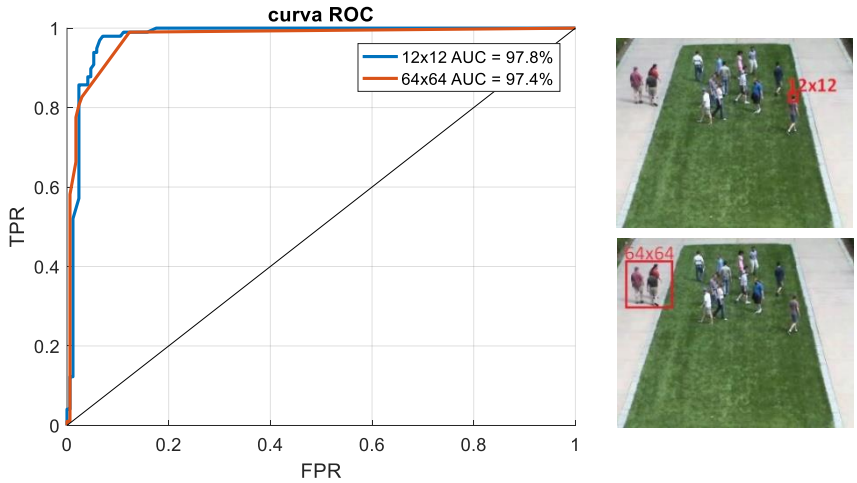


Figura 4-2: Curvas ROC óptimas UMN 1 con KM

b) Fuzzy C-Means

Una vez estudiado el funcionamiento del sistema con el algoritmo KM, se va a evaluar la implementación de FCM propuesta para ver si se consigue mejorar los resultados obtenidos hasta ahora y si, como se desea, este algoritmo es capaz de modelar mejor el comportamiento de la escena; o si por el contrario la complejidad introducida hace también que el rendimiento se vea empobrecido. Se procede de la misma manera que se hizo anteriormente, comenzando con la evaluación de las características para encontrar el vector de características que mejor describa la escena y después fijar los parámetros del algoritmo.

Con el número de *clusters* fijado al mínimo (2 en este caso, consultar apartado 3.1.2 y Anexo C), se calcula la Figura 0-7, resultante de la ejecución del detector con la característica del gradiente temporal G_t. En esta ocasión, el gradiente temporal es mucho menos discriminante, pero como se conoce del análisis previo que la característica sí es un buen descriptor, se analiza conjuntamente con el número de clusters y se muestra en Figura 0-14.

A continuación, se evalúa ahora el desempeño de las dos características de tasas de cruces por cero, $[ZCR_x, ZCR_y]$, con las que se obtienen los resultados de la Figura 0-8. Como el gradiente temporal arroja buenos resultados, se espera que esta ejecución no mejore dichos resultados, y en efecto así se produce aunque sí que aporta otra información interesante.Para tamaños pequeños no aporta información útil pues está en torno a 0.5, pero sí presenta dos grandes máximos cuando el tamaño de cuboide crece. Es especialmente relevante el máximo centrado en 132x132 píxeles, que tiene sentido pues con ese tamaño se puede incluir en un mismo cuboide centrado el grupo más denso de las personas en movimiento presentes. Además, para los mayores tamaños analizados se demuestra que se tiene un muy buen rendimiento, pues engloban en un único cuboide todo el movimiento de la escena y son capaces de detectar las anomalías de manera global.

Se pasa ahora al análisis de los gradientes espaciales, comenzando con $[G_x, G_y]$, con los que el rendimiento obtenido es el que la Figura 0-9 refleja. En esta ocasión, como ocurrió para KM, se tienen picos muy altos de rendimiento pero el aspecto de la curva es muy ruidoso, por lo que puede intuirse que la información es útil pero posiblemente pueda ser mejor representada. La gráfica parece mostrar tres picos de rendimiento distribuidos a lo largo del eje X, siendo el primero de ellos el más alto pero también el que mayor variabilidad presenta en su entorno.

Se analiza a continuación el sistema con el empleo del HOG para ver es capaz de solucionar este problema, y se refleja en la Figura 0-10. Esta curva, si bien no elimina todo lo deseable la variabilidad dependiente del tamaño de cuboide, sí aumenta el AUC para casi todos los tamaños. La distribución de los picos cambia respecto a los gradientes, por lo que se va a probar a juntar ambas características para ver si pueden complementarse y se consigue una mejora global. Esto se muestra en la Figura 0-11 y permite observar que en efecto se puede conseguir una curva con la envolvente mucho más plana para todo el eje de cuboides pero ni se alcanzan los picos de AUC obtenidos por separado, ni se consigue eliminar la variabilidad existente sino todo lo contrario. De esta manera, se descarta el uso conjunto de ambas características.

Para tratar de mejorar el funcionamiento de este detector mediante FCM se va a intentar encontrar características que puedan emplearse juntas para lograr clasificar mejor las anomalías presentes en la escena. Para ello, y dado que por separado son las características que mejores resultados arrojan, se elige el gradiente temporal y el HOG para crear un vector de características que tendrá tamaño 10 y producirá los resultados mostrados en la Figura 0-12.

A continuación, se procede, de igual manera que para KM, al estudio del parámetro de número de *clusters* para el algoritmo FCM. El rendimiento según aumenta dicho número se muestra en la Figura 0-15, y de ella se puede extraer que para 3 los resultados mejoran respecto a 2, y que más allá no solo no se consigue una mejora sino que la oscilación de los mismos crece (no se muestran más en el gráfico para mantener la claridad del mismo).

El pico de rendimiento máximo del sistema está por tanto, empleando HOG y gradiente temporal, en el tamaño de cuboide de 108x108 píxeles, y se tienen otros picos pero de notable menor altura, por lo que este será el tamaño elegido. El buen funcionamiento del mismo se

justifica porque es un tamaño con en el que puede englobar prácticamente a la gran mayoría de las personas en movimiento; como se muestra en la Figura 4-3 junto con la curvas ROC correspondiente.

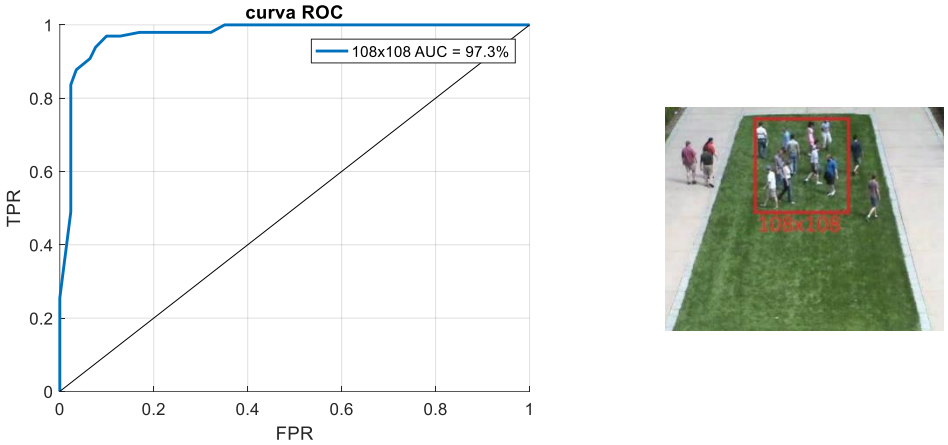


Figura 4-3: Curvas ROC óptimas UMN 1 con FCM

Como se puede apreciar, se alcanza un rendimiento superior al del 97%, muy bueno, pero también se genera una dependencia del tamaño de cuboide elegido mucho mayor y de aspecto menos uniforme. En este caso no se ha conseguido mejorar a KM, se piensa que debido al color del fondo de la escena que provoca una peor detección de los agentes cuando se tiene en cuenta la existencia de dispersión.

Escena 2

En esta segunda escena la anomalía es del mismo carácter que en la anterior y en la que se verá después, y se produce cuando el grupo de personas en movimiento comienza a correr. Respecto a la Escena 1, esta tiene una deformación por perspectiva ligeramente mayor y, como más relevante, presenta unos cambios de iluminación grandes cuando se abren y cierran las puertas situadas a la izquierda de la imagen. Siguiendo el protocolo descrito, y apoyándose en las figuras del Anexo A, se obtienen los resultados que se muestran a continuación

a) K-Means

Funciona especialmente bien el HOG, pues tiene picos realmente altos y presentan una variación baja en determinadas zonas del eje de cuboides, mientras que el gradiente temporal es más bajo pero también mucho más constante. Por ello, se decide para esta escena recurrir al empleo conjunto de ambas características, siendo el gradiente la que complementa al HOG por tener menos peso en la *clusterización*, como puede verse en la Figura 0-20 de la Tabla 0.3.

Con ello, se obtiene el pico de rendimiento del sistema para 3 *clusters*, para el tamaño de cuboide de 56x56 píxeles, aunque también son muy relevantes los tamaños de 20x20 y 148x148, quedando todos ellos por encima del 97% de rendimiento como se ve en la Figura 4-4. Ordenando estos tamaños de menor a mayor, se obtiene una detección de anomalías local por persona (o partes de persona), local por agrupación de personas, o global de todos los individuos de la escena.

En este caso, se piensa que la característica de gradiente temporal funciona peor debido a los cambios de iluminación tan bruscos que presenta la escena, y por ello se ha de recurrir, con sus ventajas e inconvenientes, al HOG.

b) Fuzzy C-Means

Al realizar el salto hacia el modelo de FCM, los resultados varían como era de esperar en cuanto al desempeño particular pero no en cuanto a la validez de las características. Se puede ver en el anexo antes citado, en la Tabla 0.4, cómo el gradiente temporal aporta menos que en la escena anterior (Figura 0-22), la tasa de cruces por cero tan sólo es ligeramente útil para tamaños grandes (Figura 0-23), y la información espacial si presenta zonas de rendimiento elevado, más altas y más estables con el HOG (Figura 0-24 y Figura 0-25)..

En este caso, como en el anterior, se emplean conjuntamente el gradiente temporal y el HOG en busca de ofrecer un mejor rendimiento (Figura 0-26) Al aumentar el número de *clusters* a 3 mejora el resultado, pero 4 no aporta ninguna mejora frente a 3. La mejor detección se hace de forma global con un tamaño de cuboide algo menor que en KM, 100x100, pues como no se está localizando la anomalía en el espacio, con este tamaño se puede encajar perfectamente al grupo más denso de personas y se detectará de forma óptima para este sistema. Como se puede ver, se mejoran los resultados hasta superar el 98.5% de rendimiento, teniendo en cuenta que el tiempo necesario para su ejecución también aumenta y que aunque este máximo sea más alto también la detección es menos estable.

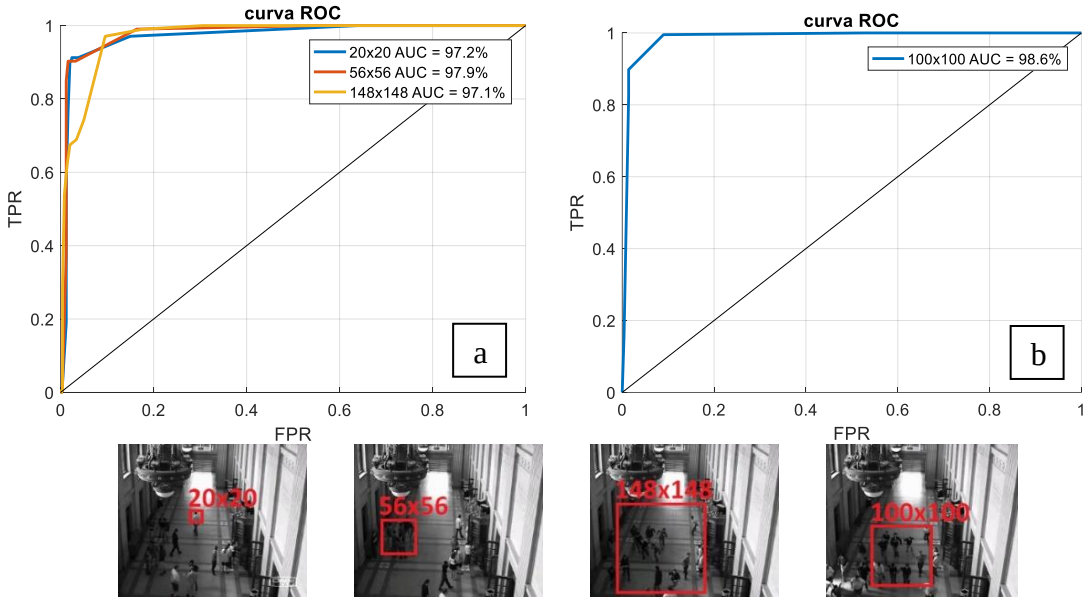


Figura 4-4: Curvas ROC óptimas UMN 2 con KM (a) y FCM (b)

Escena 3

Esta última escena del *dataset* presenta una gran similitud con la primera analizada, la perspectiva es similar y la localización de las personas también, cambian el fondo (suelo) y muy ligeramente el tamaño de las personas, por lo que en un principio cabría esperar resultados similares. Analizando las figuras mostradas en el Anexo A, obtenidas siguiendo el protocolo descrito, se establecen los siguientes resultados.

a) *K-Means*

El algoritmo definitivamente tiene un buen funcionamiento para esta secuencia (Tabla 0.5), con una curva de rendimiento alta y constante para el gradiente temporal y un gran pico del HOG para los valores bajos de tamaño de cuboide. De esta manera, se emplea el gradiente como apoyo del HOG para optimizar la detección y se calcula la curva ROC de la Figura 4-5, que muestra los resultados para los tamaños de 12x12 y 24x24 píxeles. Para estos resultados, se ha empleado una clasificación con 2 clases o *clusters*, pues mejora significativamente respecto a 1 y se mantiene constante para 3.

Como queda demostrado, en esta escena KM ofrece un rendimiento excelente como detector local de anomalías pues para los tamaños pequeños de cuboide todas las curvas están en el entorno de las aquí mostradas (96% - 97%), que se corresponden con el tamaño de personas en la imagen, ya sea completas o divididas en cabeza, tronco y piernas.

b) *Fuzzy C-Means*

Como está ocurriendo en todas las secuencias analizadas hasta el momento, FCM es capaz de ofrecer un pico de rendimiento más alto que KM como el que muestra la Figura 4-5, pero a costa del tiempo de ejecución y sobre todo de la estabilidad de los resultados, pues su dependencia con el tamaño de cuboide crece.

En la Tabla 0.6, se puede ver cómo el gradiente temporal apenas aporta información útil para 2 *clusters* pero mejora para más, y las ZCRs tan sólo funcionan para los tamaños de cuboide más altos, por lo que se recurre a HOG por tener más y mayores picos de rendimiento, con 3 como el número de *clusters* elegido tras analizar el parámetro. Se prueba además si con ZCRs se puede mejorar la solución pero esta no tiene el peso necesario para afectar a la *clusterización* posterior por lo que se desestima su uso.

FCM se presenta en este caso como un mejor detector de anomalías tratando los agrupamientos de individuos que de forma localizada con cada uno de ellos, pues se alcanza un rendimiento del 97.8% para un tamaño, 204x204 píxeles, que abarca todo el movimiento de la escena.

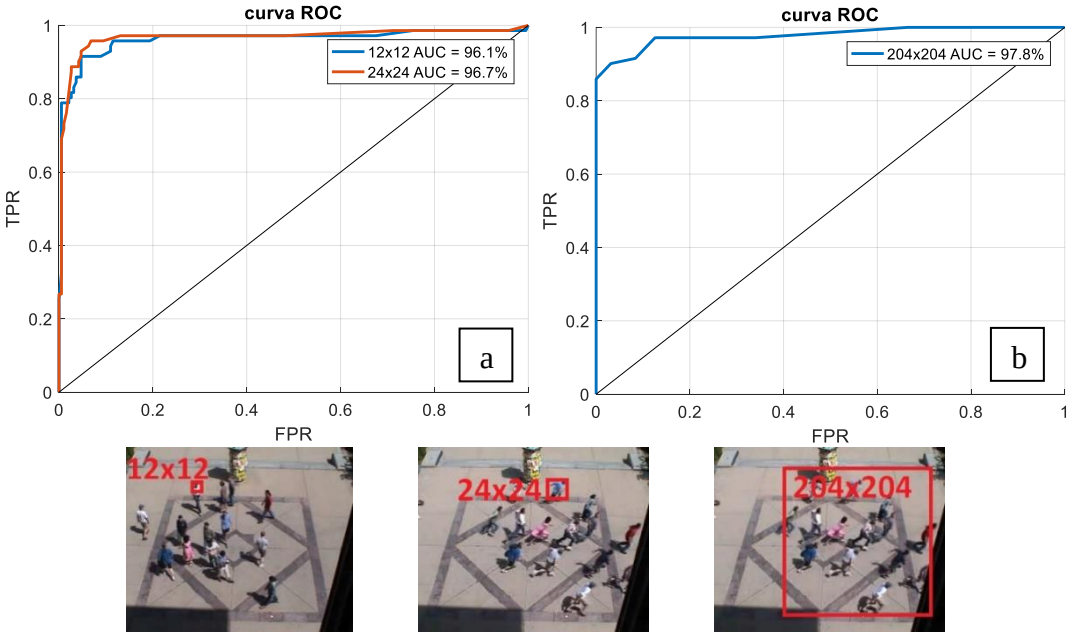


Figura 4-5: Curvas ROC óptimas UMN 3 con KM (a) y FCM (b)

Con todo lo visto hasta ahora, se puede concluir que de los dos métodos propuestos, en estas escenas KM funciona mejor como detector local de anomalías para un único *cluster*, siendo capaz de modelar el comportamiento de cada uno de los individuos presentes en la escena o de pequeños grupos (2-3 personas) cuando estos son muy densos. Esto, no obstante, está directamente relacionado con las características y el número de *clusters* empleados para la clasificación y ajustando FCM se consigue, como mínimo, igualar el rendimiento de KM.

El gradiente temporal funciona generalmente bien y para tamaños de cuboide pequeños, pues se están detectando eventos anómalos producidos por la velocidad del movimiento y la comparación de cuboides consecutivos por tanto permite una buena detección; mientras que el HOG tiende a generar unos picos de rendimiento más altos pero mucho más variables con el tamaño de cuboide, sólo funcionando bien para aquellos tamaños que comprenden al fenómeno anómalo. La tasa de cruces por cero apenas mejora ligeramente al gradiente cuándo este no es un buen descriptor de la secuencia, por lo que no se ha empleado, y los gradientes espaciales nunca se usan directamente pero ha quedado demostrada su utilidad pues dan lugar al HOG.

En [4] se muestran las AUC obtenidas para este *dataset* por las propuestas más relevantes de la literatura hasta el momento. Estas, si bien es cierto que no actúan a nivel de imagen como se hace en este trabajo sino que localizan las anomalías, fluctúan entre 0.84 y 0.99, y puesto que aquí no se pretendía tal precisión y se obtienen valores superiores a 0.97 para ambos casos, se puede afirmar que se han obtenido unos resultados muy satisfactorios para lo sencillo del sistema implementado, aunque se desearía una menor dependencia con el tamaño de cuboide.

4.4.2 Escaleras mecánicas

Las escaleras mecánicas son el objetivo final de este trabajo, por lo que se prestará especial atención a este tipo de secuencias. Este *dataset*, aunque es pequeño, incluye ejemplos de las anomalías más típicas que pueden encontrarse en este entorno, que son personas corriendo, caídas y empujones. La primera gran diferencia respecto al *dataset* estudiado antes es que mientras que la resolución de las secuencias de entrenamiento (normalidad) es de 720x480 píxeles, mientras que las de test son de 704x576, lo que obliga a hacer un pre-procesado para igualar ambas prestando atención a no modificar la relación de aspecto. Además, esta mayor resolución provocará un aumento en el tiempo de procesamiento tanto de extracción de características como de *clusterización* de las mismas, pues la cantidad de cuboides, y por tanto el volumen de datos, crece.

Dentro de este *dataset*, todas las secuencias muestran dos escaleras, una de subida y una de bajada y una densidad de personas que va desde cero (la mayoría del tiempo) hasta alcanzar la decena de individuos, sin embargo se incluyen secuencias con una vista desde la base de las escaleras mecánicas y secuencias con vista superior que cambiarán totalmente la perspectiva, como se muestra en la Figura 4-6.

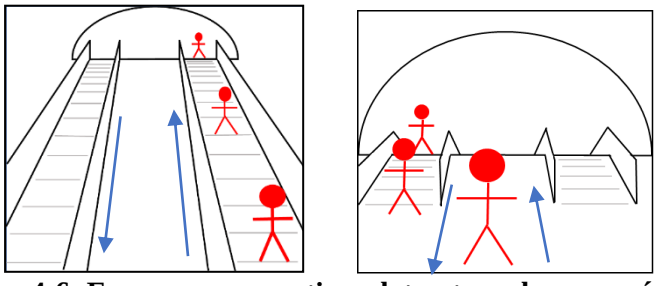


Figura 4-6: Esquema perspectivas dataset escaleras mecánicas

- ***Anomalous_run_left***

En esta secuencia la anomalía a detectar es un hombre que corre hacia abajo por la escalera izquierda. Recorre toda la escalera, por lo que al inicio el tamaño de la persona en la escena es mucho menor que al final de la misma. El entrenamiento se realiza con una secuencia de normalidad en la que dos grupos de personas bajan por el movimiento de las escaleras sin cometer ninguna anomalía.

a) *K-Means*

Con el primer algoritmo implementado se obtienen los resultados que se muestran en la Tabla 0.7 para esta secuencia. En ella, se puede ver cómo para las características de cruces por cero, gradientes espaciales y HOG apenas aportan información útil, pues las dos primeras se mantienen muy constantes entorno al valor de 0.5 y el HOG aunque tenga una envolvente superior algo mayor es muy poco estable. No obstante, el gradiente temporal tiene un desempeño no brillante pero sí correcto, formando una curva descendiente según se aumenta el tamaño de cuboide. Podría plantearse en este caso, puesto que parece claro emplear el gradiente temporal y tamaños pequeños, juntar esta característica al HOG para complementarlo, pero ya se ha demostrado que los 9 intervalos que introduciría harían que el gradiente pierda peso y aumente el número de errores. El mejor rendimiento se obtiene, de esta manera, para tamaños de cuboide de entre 12x12 y 44x44 píxeles, siendo el más pequeño de ellos el que más AUC tiene y el que genera la Figura 4-7.

b) *Fuzzy C-Means*

A la vista de los resultados mostrados en la Tabla 0.8, la detección por medio de FCM parece pseudo-aleatoria, pues las características temporales apenas superan el 0.5 de AUC y las espaciales se muestran muy ruidosas. No obstante, como se conoce del análisis de KM que para esta escena el gradiente temporal puede ser muy discriminante, se opta por seleccionar esta característica y aumentar el número de *clusters* empleados, logrando igualar el comportamiento del detector con este algoritmo y característica con el de KM e incluso mejorarlo. Se muestran conjuntamente los resultados en la Figura 4-7, también para el tamaño mínimo de 12x12 píxeles.

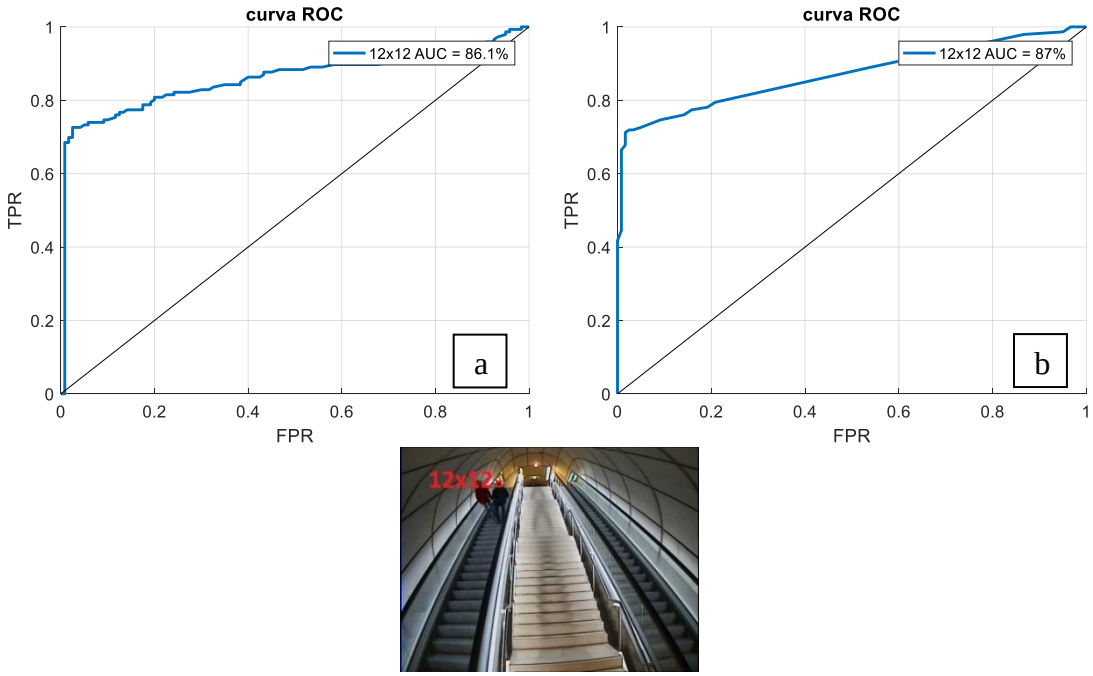


Figura 4-7: Curva ROC óptima para *anomalous_run_left* con KM (a) y FCM (b)

- ***Anomalous_push_left***

Al ser el mismo tipo vista y secuencia que el caso anterior, no es necesario realizar un nuevo entrenamiento. La anomalía que se presenta, sin embargo, es cualitativa y cuantitativamente distinta pues en esta ocasión es una persona que empuja a un maniquí haciendo que este caiga de golpe hacia delante y vaya escaleras abajo. De esta manera, existe movimiento tanto anómalo tanto prolongado en el tiempo como instantáneo localizado, por lo que todas las características pueden a priori generar algún tipo de información relevante para su detección.

a) *K-Means*

Para esta segunda secuencia de escaleras mecánicas de vista inferior, el rendimiento es similar al obtenido anteriormente con en *anomalous_run_left* con KM en cuanto a la forma que presentan las curvas de las distintas características (Tabla 0.9). Como antes, sólo el gradiente temporal es capaz de hacer una descripción correcta de la escena para los tamaños de cuboide más pequeños y además, por ser esta anomalía más pequeña tanto en el tiempo que dura como en el espacio que ocupa de la imagen, el AUC es sensiblemente menor (el peor de todos los que se obtendrán a lo largo de las pruebas).

Como puede verse en la Figura 4-8 el rendimiento ha caído hasta el 75%, que sigue siendo un rendimiento por encima de la detección aleatorio pero se considera inaceptable para un campo con aplicaciones tan críticas como la videovigilancia. Se ha elegido por su mejor rendimiento el tamaño de cuboide mínimo admitido, 12x12 píxeles, y la introducción de nuevos *clusters* no ha demostrado aportar nada por lo que se ha seleccionado sólo 1.

b) *Fuzzy C-Means*

Ocurre el mismo fenómeno que en el estudio anterior de FCM. Las curvas del análisis por características de la Tabla 0.10 no muestran resultados satisfactorios, pero se recurre al gradiente temporal y a aumentar el número de clases del espacio de características para lograr ajustar el sistema y lograr un rendimiento similar a KM, como se muestra en la Figura 4-8.

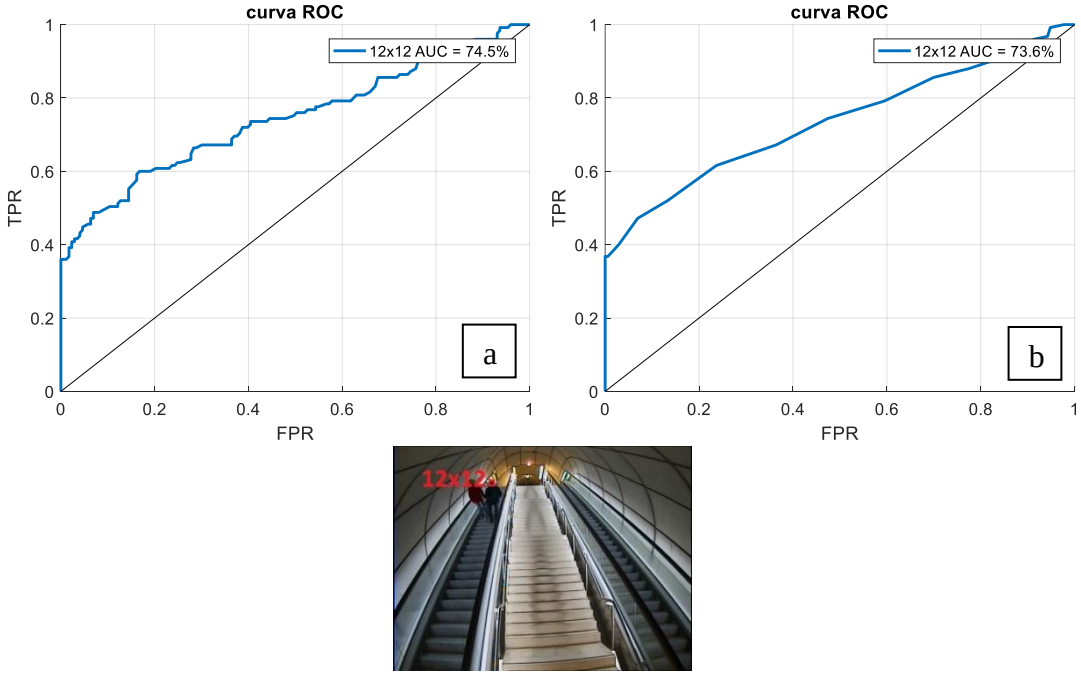


Figura 4-8: Curva ROC óptima para *anomalous_push_left* con KM (a) y FCM (b)

Los peores resultados de esta escena se justifican porque la anomalía es más corta temporalmente y más pequeña espacialmente en la escena, siendo por tanto también más difícil de detectar.

- ***Anomalous_push_up***

La anomalía de esta escena es la misma que la de la que se acaba de analizar, pero cambiando la perspectiva a un punto de vista superior. En la zona final de la escalera, una persona empuja a un maniquí, que se desploma hacia delante súbitamente como consecuencia. Debido a la perspectiva, cuando el maniquí cae no se produce movimiento significativo del agente en ninguna dirección, sino que lo que ocurre es que éste se hace más pequeño en la imagen.

a) *K-Means*

En este caso la información de gradientes espaciales y HOG no aportará información útil y serán las características temporales las que mejor permitan detectar la anomalía. Los mejores resultados se han obtenido con el uso del gradiente temporal, y estos no sólo son notablemente aceptables sino que además son muy constantes y poco dependientes del tamaño de cuboide elegido como puede verse en la Tabla 0.11.

El buen funcionamiento del sistema se debe a que es un evento muy bien localizado en el espacio y muy contundente, los agentes no cambian de tamaño en la escena y la caída se produce arriba de las escaleras, sin oclusiones y en un primer plano de la imagen. Se ha elegido el tamaño de cuboide de 136x136 píxeles y 1 *cluster* por ser el pico más alto y no aportar ninguna mejora el añadir nuevas clases, pero se supera el 90% de rendimiento para la gran mayoría de posibles casos.

b) *Fuzzy C-Means*

Es necesario también en este caso, pues las curvas de análisis independiente de características con los mínimos *clusters* no permiten inferir nada, recurrir a la característica que ya se ha demostrado para KM que funciona bien, el gradiente temporal, y tomarla como base para analizar el número de *clusters* empleados y optimizar el detector. Los resultados se muestran en la Tabla 0.12, y se selecciona el tamaño de 12x12 para 3 clases.

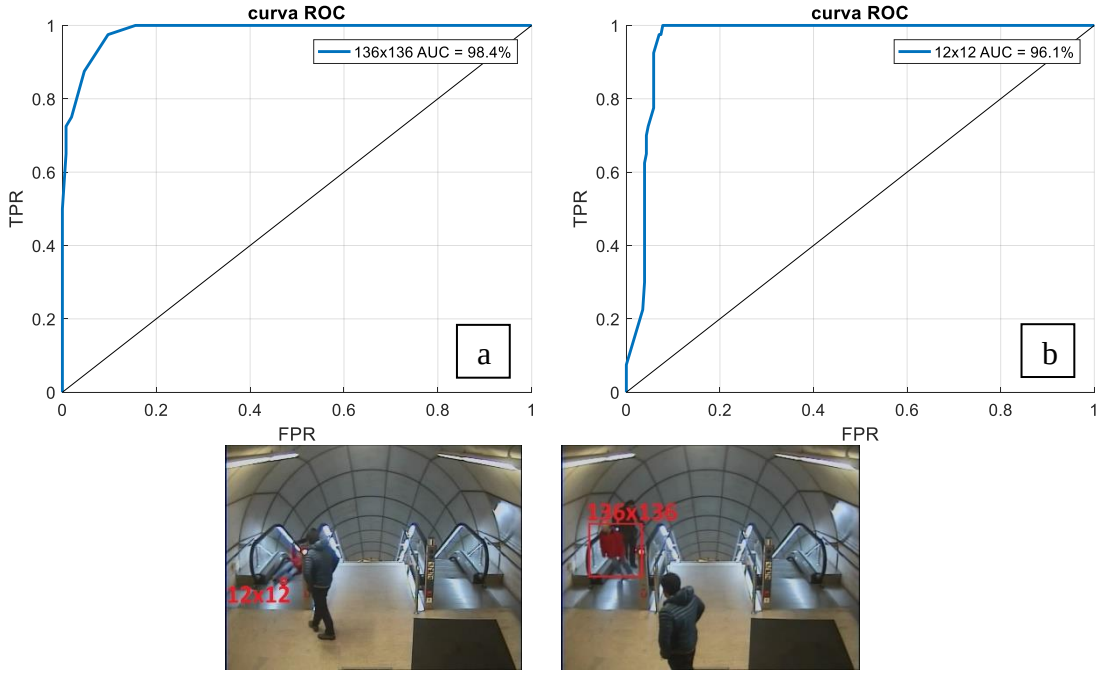


Figura 4-9: Curva ROC óptima para *anomalous_push_up* con KM (a) y FCM (b)

Como puede apreciarse este punto de vista permite una detección mucho más precisa, pero también se pierde en visibilidad global de la escalera pues sólo se ve el final.

- ***Anomalous_fall_up***

Esta es la última de las escenas analizadas. Es muy similar a la anterior y, por ser de la misma vista, no es necesario realizar un nuevo entrenamiento del sistema. En ella la anomalía también es un maniquí que cae, pero en este caso no es motivado por un empujón sino por su propio peso lo que hace que la caída sea hacia delante y, junto con el fenómeno explicado de que pase a ocupar menos píxeles en la imagen por la perspectiva, también se produce un pequeño movimiento en el sentido de la caída.

a) *K-Means*

La Tabla 0.13 muestra los resultados obtenidos, entre los que destaca una vez más el gradiente temporal, que será muy estable y poco dependiente del tamaño de cuboide elegido. A pesar de ello, la curva presenta una menor altura para todo el eje de cuboides debido a que el movimiento en este caso es más gradual y por tanto el gradiente temporal entre los cuboides consecutivos será menor y su detección será más difícil.

De todos modos, como se ve en la Figura 4-10 se siguen teniendo rendimientos superiores al 90% por lo que puede concluirse que la detección es buena Se muestran los resultados de los tamaños de 12x12 píxeles y 100x100, de nuevo para 1 único *cluster* al no manifestarse un beneficio por introducir un mayor número de ellos.

b) *Fuzzy C-Means*

Como en todos los casos, ajustando el número de *clusters* a 3 y empleando el gradiente temporal se obtiene un rendimiento superior al 90% para los tamaños pequeños de cuboide, como se puede ver en la Figura 4-10. Los resultados se muestran en la Tabla 0.14 y son tan buenos y constantes como los de KM. Para el tamaño de 12x12 el rendimiento supera a KM, pero no se alcanza el valor máximo de AUC mediante este algoritmo.

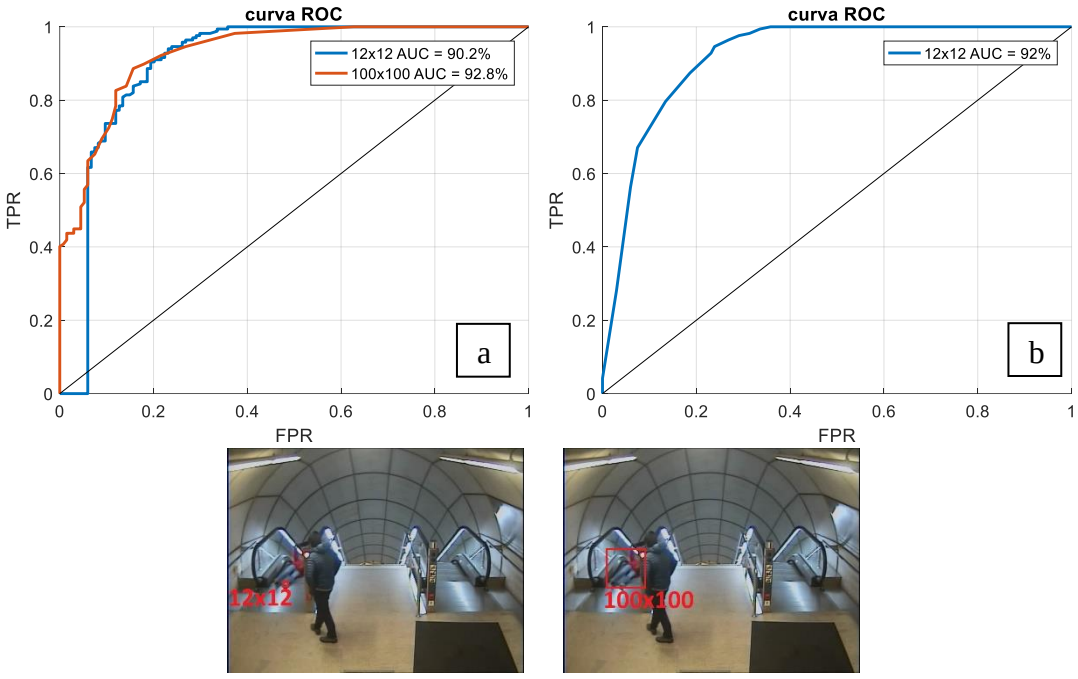


Figura 4-10: Curvas ROC óptimas *Anomalous_fall_up* con KM (a) y FCM (b)

Queda de esta manera demostrado que se han conseguido unos buenos resultados de rendimiento para las secuencias de escaleras empleando tanto KM como FCM como algoritmos de clasificación de anomalías, y en este caso la característica más determinante ha sido el gradiente temporal. Los peores resultados se han obtenido en la secuencia *anomalous_run_left* debido a las limitaciones del empleo de gradientes y las características de la escena. Salvo la excepción en que FCM tan solo ha logrado a igualar a KM, este ha mostrado un mejor comportamiento siempre y cuando se pueda admitir la carga computacional que implica y la posible mayor variabilidad de los resultados en cuanto crece el tamaño de cuboide.

Para tamaños de cuboide pequeños, ambos métodos junto con el gradiente temporal arrojan resultados fiables y estables, mientras que si se emplean el HOG junto con tamaños más grandes de cuboide, que comprendan la anomalía al completo, pueden mejorarse los sensiblemente los resultados. Por ello, si se desea abstraerse del tamaño del cuboide a la hora de entrenar el sistema, se optará por un 12x12 junto con gradientes temporales, que aseguran siempre un funcionamiento correcto, mientras que si se puede ajustar en detalle para cada escena el sistema será recomendable incorporar también los HOGs y adaptarse al tamaño de las anomalías.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

La gran y primera conclusión que puede extraerse de la realización del trabajo es que se han cumplido los requisitos marcados al inicio. Se ha conseguido construir un sistema completo detector de anomalías y, teniendo en cuenta lo sencillo de las características y algoritmos empleados, se han logrado unas tasas de acierto notablemente altas.

En cuanto a las características extraídas, se ha demostrado que las más descriptivas, en función del tipo de secuencia analizada y el algoritmo utilizado, han sido el gradiente temporal y el HOG siendo la primera la característica de mayor estabilidad y la segunda la de mayores picos de rendimiento. Para las secuencias de escaleras ha funcionado especialmente bien el gradiente temporal, mientras que para el *dataset* UMN ha sido muy empleado también el HOG. Esto se debe a la mayor resolución de las secuencias de escaleras, que permite tener una mayor precisión en los cuboides y que los gradientes temporales tengan por tanto una información precisa (el tamaño más pequeño, 12x12 píxeles, es el que mejor resultado produce con el gradiente temporal), frente a las secuencias con menor resolución en las que no se puede tener tanta precisión con los agentes y por ello los HOGs, que describen el comportamiento local de los agentes que engloban los cuboides, funcionan mejor. Además, los cambios de tamaño de los agentes en la escena por la perspectiva son imperceptibles en UMN pero muy notables en escaleras mecánicas, lo que hace que la característica del HOG sea adecuada, como se ha visto, para el primero de los *datasets*, pero apenas tenga utilidad en el segundo.

La característica de tasa de cruces por cero propuesta nunca ha llegado a lograr tanto acierto como el gradiente temporal cuando éste ha funcionado bien, pero para los casos en que no ha aportado información útil si ha logrado su objetivo. Los gradientes espaciales se ha demostrado que son útiles para generar el HOG, pero no son capaces de modelar los eventos con la precisión necesaria para este trabajo. Por último, es relevante también que los 9 intervalos del HOG hacen que, cuando este se emplea, las otras características pierdan peso en la *clusterización* y su impacto en la detección final se ve muy reducido.

Respecto a los algoritmos empleados, se podría decir que definitivamente la implementación realizada del algoritmo KM es muy satisfactoria, pues de una manera rápida y sencilla es capaz de ofrecer unas buenas tasas de aciertos para las secuencias empleadas. FCM logra en la mayoría de los casos mejorar levemente los resultados de KM, pero a costa de un tiempo de ejecución mucho mayor, por lo que deberá asegurarse una buena capacidad de cómputo del sistema si se emplea este último algoritmo. Si esto no es un problema como en este caso, por lo sencillo que es el sistema y lo pequeño que es el vector de características, se recurrirá por tanto a FCM.

De esta manera, se proponen diferentes configuraciones finales del sistema. Si se busca la mayor tasa de aciertos se propone hacer un análisis detallado previo de la escena a analizar con secuencias de entrenamiento de normalidad y un *Ground-Truth* que permita, como se ha hecho en este trabajo, decidir qué características (probablemente HOG pues se ha visto que al ajustarlo ofrece mayor AUC que las demás), algoritmos y parámetros se ajustan mejor a la escena; se selecciona el pico de rendimiento más alto obtenido y se fija el sistema para cada escena en particular. Por el contrario si se busca un sistema con características y algoritmo de modelado fijos, de manera que sólo sea necesario un nuevo entrenamiento para poder analizar una escena diferente, se tomarán cuboides lo más pequeños posibles, en este caso 12x12 píxeles, y se analizará la característica de gradiente temporal mediante FCM

definido con 3 *clusters*. Como último caso, si se necesita una complejidad computacional baja y una ejecución muy rápida del algoritmo se tomará KM con un vector de características de tamaño 1, que será el gradiente temporal por haberse demostrado su utilidad.

5.2 Trabajo futuro

Dada la limitación, tanto en tiempo como en espacio, que ha de tener este Trabajo Fin de Grado, no se puede llevar a cabo todas las implementaciones y pruebas que se desearían y que se han llegado a plantear. Por ello, se proponen en este apartado diversas líneas de trabajo futuras para, partiendo de este trabajo, llevar a cabo un sistema que mejore el funcionamiento del aquí construido. Estas líneas se dividen en tres campos principales, la extracción de características, el algoritmo para modelar la escena y el análisis de los resultados.

Respecto a la extracción de características se propone añadir:

- Tamaño variable de cuboide: Esto se considera muy importante pues, debido a la perspectiva que tienen las escenas los eventos locales que en ellas suceden tienen distinto tamaño en función de dónde se localicen y, si se consigue adaptar la división en cuboides del cuadro a este fenómeno, se mejorará sensiblemente el posterior modelado.
- Otro tipo de características y filtrado de las mismas: Eligiendo otras características de las estudiadas en el estado del arte se podrá modelar de mejor o peor manera a lo que aquí se ha hecho. En este trabajo se han utilizado los gradientes de intensidad porque se deseaba que fueran características sencillas, pero si se sustituye el bloque extractor por otro más complejo el sistema seguirá siendo funcional. Se puede intuir que las características que mejor funcionarán serán las basadas en movimiento. Además, se propone realizar algún tipo de filtrado de las muestras más complejo del que se hace en este trabajo, pues la señal de imagen y por tanto de vídeo es susceptible al ruido.

En relación al algoritmo de modelado de la escena y decisión sobre anomalías, se proponen mejoras en dos líneas principales:

- La principal propuesta es la implementación de un GMM, pues se piensa que logrará clasificar mejor el contenido de la escena al introducir información de probabilidad.
- Adicionalmente, se propone cambiar el método de decisión para los implementados KM y FCM, probando otro tipo de distancias como las que se explican en el Anexo D para calcular nuevos umbrales que mejoren la detección, y estudiar el parámetro *m* del algoritmo FCM, pues se piensa que es la principal razón por la que el funcionamiento no es tan bueno como el esperado.
- Una buena mejora sería la implementación en un lenguaje de programación más rápido y eficiente que Matlab, como C++ o Python, del bloque de análisis de características y decisión para mejorar los tiempos de ejecución.

Por último, en cuanto al análisis de resultados:

- Realizar el estudio de rendimiento para otra base de datos pública, UCSD, para poder comparar mejor el rendimiento del sistema aquí propuesto con los demás desarrollados hasta la fecha.
- Evolucionar del análisis a nivel de cuadro al nivel de píxel. Se ha de ser consciente de que esto empeorará las tasas de acierto obtenidas a nivel de cuadro, pero permitirá localizar en la imagen las anomalías que se detectan y sobre todo verificar que las anomalías encontradas realmente se corresponden con las anomalías presentes en la escena.

Referencias

[1] A. A. Sodemann, M. P. Ross y B. J. Borghetti, «A Review of Anomaly Detection in Automated Surveillance,» *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, nº 6, pp. 1257-1272, 2012.

[2] T. Li, H. Chang, M. Wang, B. Ni, R. Hong y S. Yan, «Crowded Scene Analysis: A Survey,» *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 25, nº 3, pp. 367-386, 2015.

[3] O. P. Popoola y K. Wang, «Video-Based Abnormal Human Behavior Recognition-A Review,» *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, nº 6, pp. 865-878, 2012.

[4] E. B. Ermis, V. Saligrama, P. Jodoin y J. Konrad, «Abnormal behavior detection and behavior matching for networked cameras,» de *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, Stanford, CA, 2008.

[5] B.B.Orten, A. A. Alatan y T. Ciloglu, «Event detection in automated surveillance systems,» de *2006 IEEE 14th Signal Processing and Communications Applications*, Antalya, 2006.

[6] T. V. Duong, H. H. Bui, D. Q. Phung y S. Venkatesh, «Activity recognition and abnormality detection with the switching hidden semi-Markov model,» de *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),*, San Diego, CA, 2005.

[7] T. Xiang y S. Gong, «Scene event recognition without tracking. Special issue on visual surveillance of dynamic scenes,» *Acta Automatica Sinica (Chinese Journal of Automation), Chinese Academy of Sciences,* vol. 29, pp. 321-331, 2003.

[8] A. Singh, S. Sawan, M. Hanmandlu, V. Madasu y B. Lovell, «An abandoned object detection system based on dual background segmentation,» de *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, Genova, 2009.

[9] T. Cao, X. Wu, J. Guo, S. Yu y Y. Xu, «Abnormal crowd motion analysis,» de *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, 2009.

[10] T. Hayashi y K. Yamada, «Predicting unusual right-turn driving behavior at intersection,» de *2009 IEEE Intelligent Vehicles Symposium*, Xi'an, 2009.

[11] A. Mishra, K. Sudan y H. Soliman, «Detecting border intrusion using wireless sensor network and artificial neural network,» de *2010 6th IEEE International Conference on Distributed Computing in Sensor Systems Workshops (DCOSSW)*, Santa Barbara, CA, 2010.

[12] N. H. C. Yung y A. H. S. Lai, «An effective video analysis method for detecting red light runners,» *IEEE Transactions on Vehicular Technology*, vol. 50, nº 4, pp. 1074-1084, 2001.

[13] A. Panangadan, M. Mataric y G. Sukhatme, «Detecting anomalous human interactions using laser range-finders,» de *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, Sendai, Japan, 2004.

[14] J. Kim y K. Grauman, «Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates,» de *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2010.

[15] Z. Chen, «Video Anomaly Detection Based on Local Statistical Aggregates,» de *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012.

[16] M. Sonka, V. Hlavac y R. Boyle, Image Processing, Analysis, and Machine Vision, Cengage Learning, 2014, pp. Cap. 5.3, 9.5.

[17] P.-M. Jodoin, Y. Benezeth y Y. Wang, «Meta-tracking for video scene understanding,» de *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Krakow, 2013.

[18] B. Zhou, X. Wang y X. Tang, «Random field topic model for semantic region analysis in crowded scenes from tracklets,» de *CVPR 2011*, Providence, RI, 2011.

[19] I. Steinberg, T. M. London y D. D. Castro, «Hand Gesture Recognition in Images and Video,» *CCIT Report*, nº 763, 2010.

[20] V. Chandola, A. Banerjee y V. Kumar, «Anomaly Detection: A Survey,» *ACM Computing Surveys*, vol. 41, nº 3, 15, 2009.

[21] M. Charytanowicz, J. Niewczas, P. A. Kowalski, P. Kulczycki, S. Łukasik y S. Zak, «Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images,» de *2nd International Conference on Information Technologies in Biomedicine*, Kamien Slaski, 2010.

[22] J. Irani, N. Pise y M. Phatak, «Clustering Techniques and the Similarity Measures used in Clustering: A Survey,» *International Journal of Computer Applications*, vol. 134, nº 7, pp. 9-14, 2016.

[23] L. Seidenari, «Supervised and Semi-supervised Event Detection with Local Spatio-Temporal Features,» Dottorato, Universit`a degli Studi di Firenze, 2005.

[24] N. Dalal y B. Triggs, «Histograms of Oriented Gradients for Human Detection,» de *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, 2005.

[25] D. Arthur y S. Vassilvitskii, «k-means++: The Advantages of Careful Seeding,» de *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, New Orleans, LO, 2007.

[26] Z. Cebeci y F. Yildiz, «Comparison of K-Means and Fuzzy C-Means Algorithms on Different Cluster Structures,» *Journal of Agricultural Informatics*, vol. 6, nº 3, pp. 13-23, 2015.

Anexos

A Figuras de análisis

Tabla 0.1 Rendimiento UMN 1 con KM

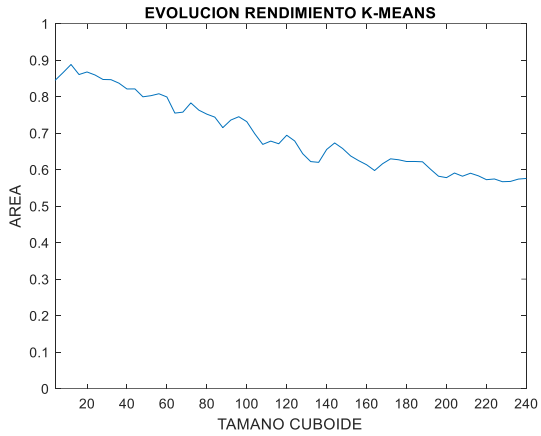


Figura 0-1: Rendimiento UMN 1 con KM y G_t

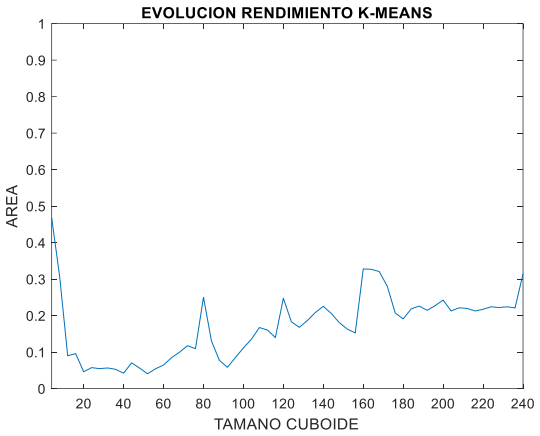


Figura 0-2: Rendimiento UMN 1 con KM y $ZCR_{x,y}$

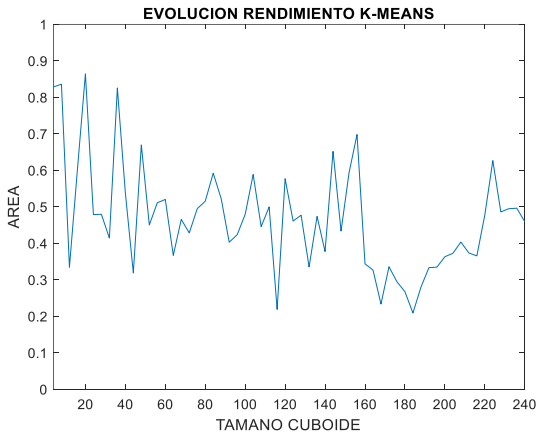


Figura 0-3: Rendimiento UMN 1 con KM y $G_{x,y}$

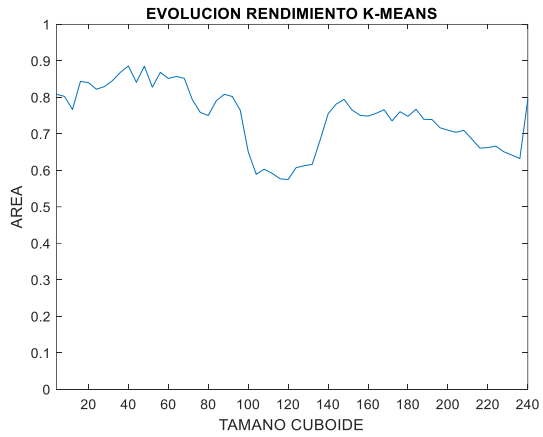


Figura 0-4: Rendimiento UMN 1 con KM y HOG

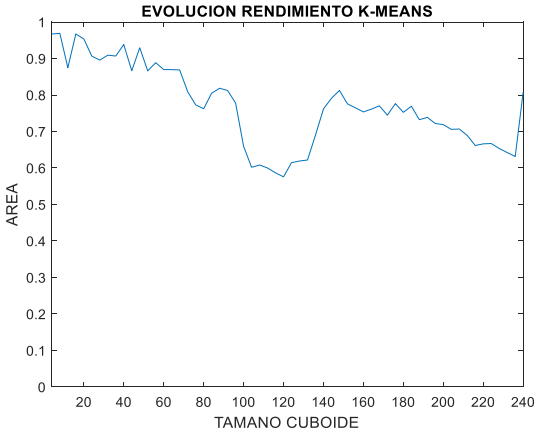


Figura 0-5: Rendimiento UMN 1 con KM y $[G_t, HOG]$

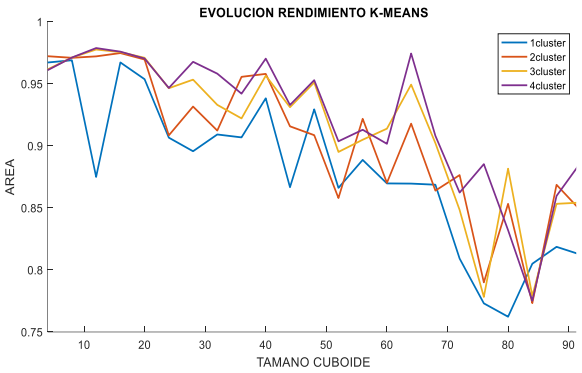


Figura 0-6: Comparativa Clusters UMN 1 con KM y $[G_t, HOG]$

Tabla 0.2 Rendimiento UMN 1 con FCM

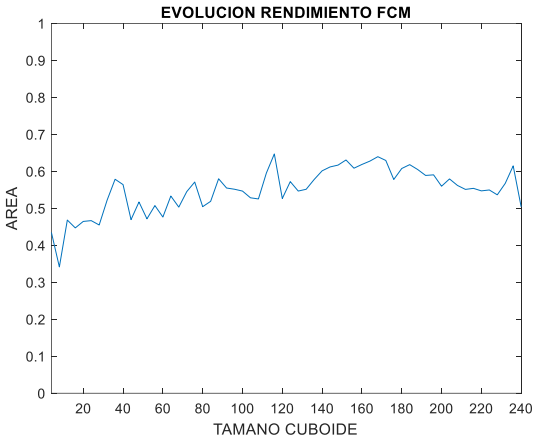


Figura 0-7: Rendimiento UMN 1 con FCM y G_t

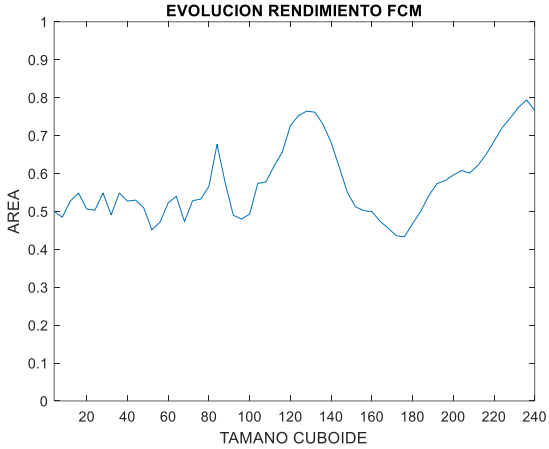


Figura 0-8: Rendimiento UMN 1 con FCM y $ZCR_{x,y}$

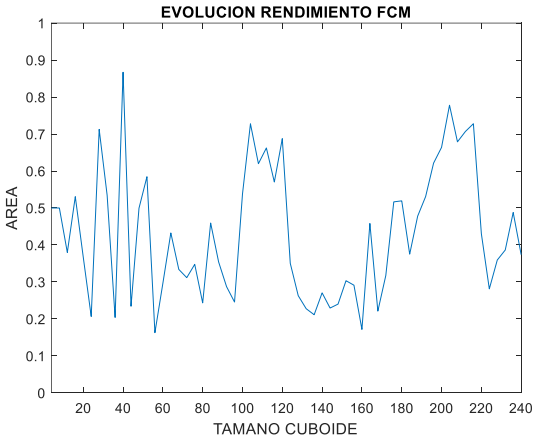


Figura 0-9: Rendimiento UMN 1 con FCM y $G_{x,y}$

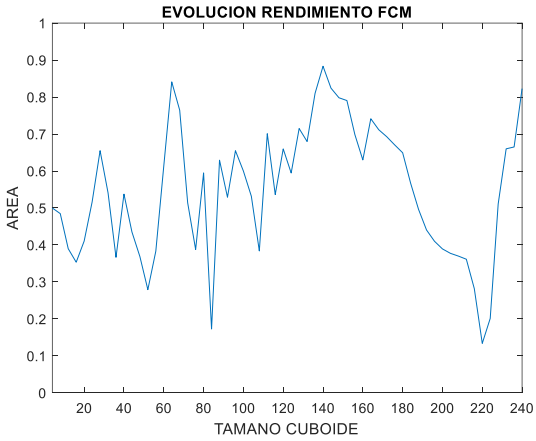


Figura 0-10: Rendimiento UMN 1 con FCM y HOG

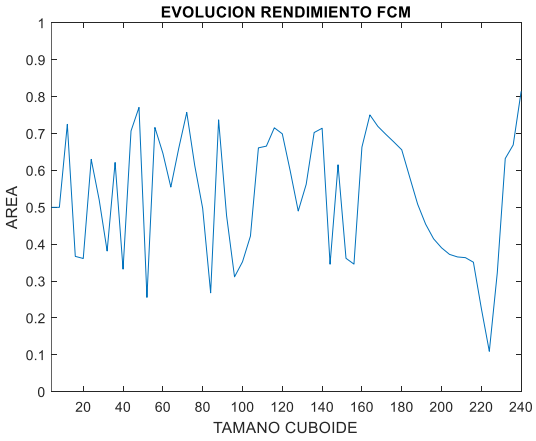


Figura 0-11: Rendimiento UMN 1 con FCM y $[G_{x,y}, HOG]$

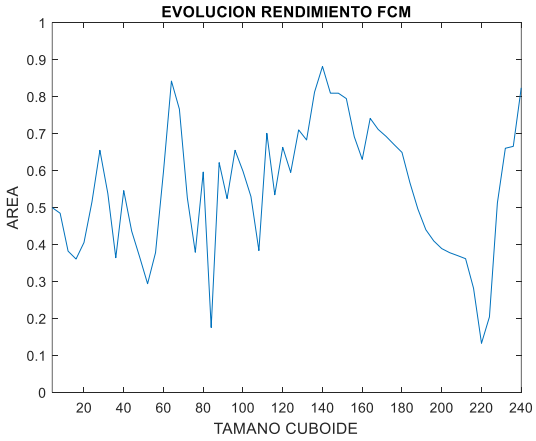


Figura 0-12: Rendimiento UMN 1 con FCM y $[G_t, HOG]$

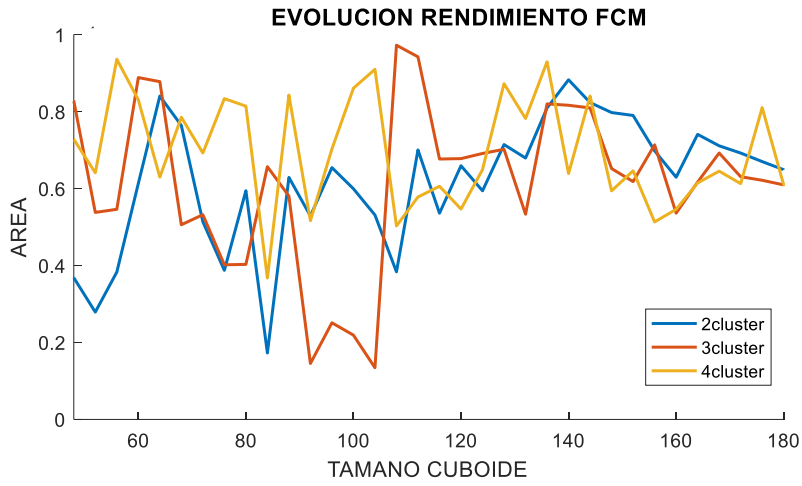


Figura 0-13: Comparativa Clusters UMN 1 con FCM y HOG

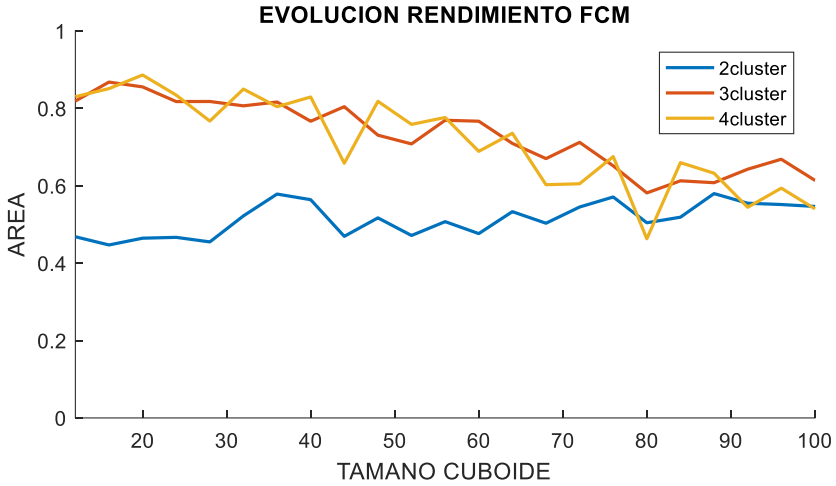


Figura 0-14: Comparativa Clusters UMN 1 con FCM y Gt

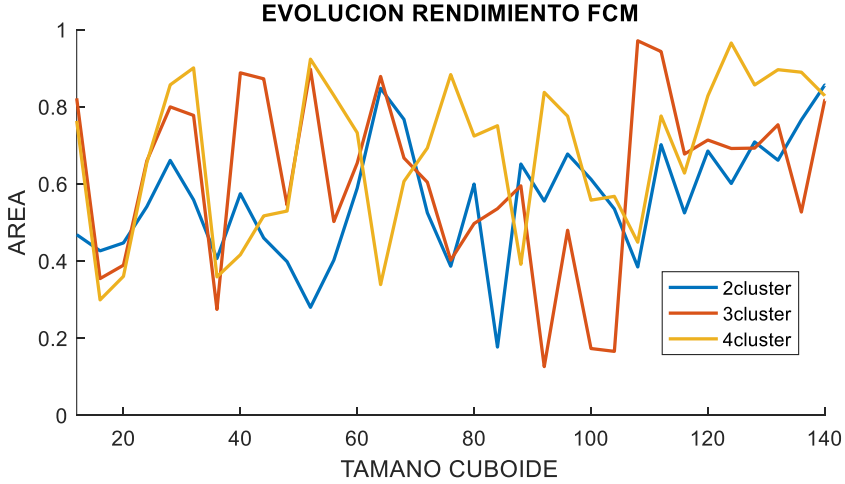


Figura 0-15: Comparativa Clusters UMN 1 con FCM y $[Gt, HOG]$

Tabla 0.3 Rendimiento UMN 2 con KM

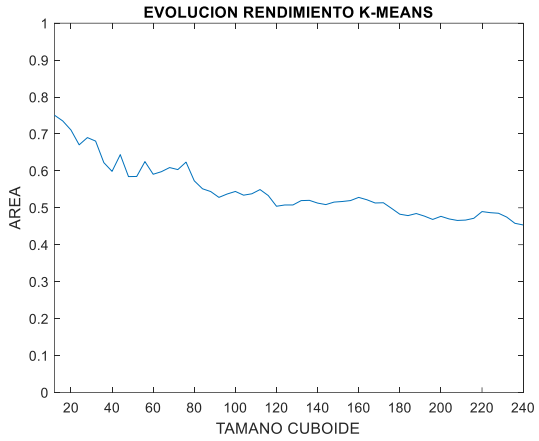


Figura 0-16: Rendimiento UMN 2 con KM y G_t

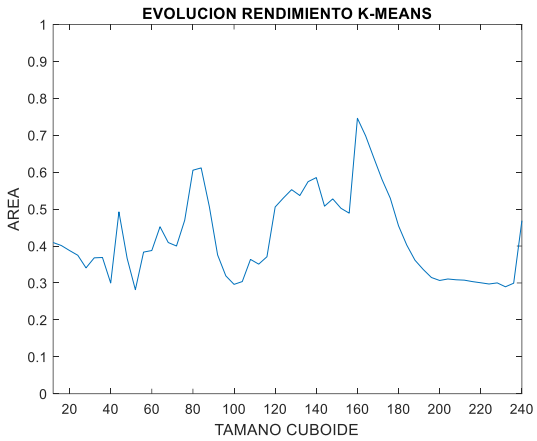


Figura 0-17: Rendimiento UMN 2 con KM y $ZCR_{x,y}$

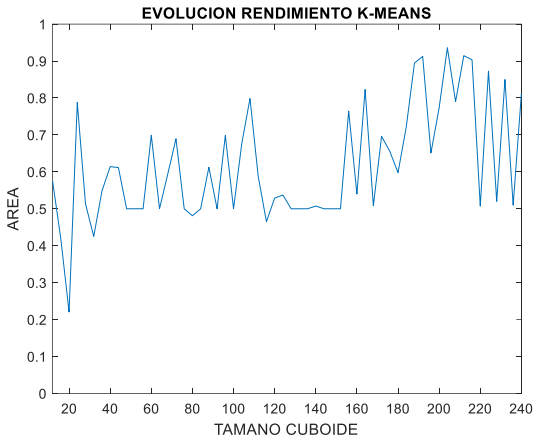


Figura 0-18: Rendimiento UMN 2 con KM y $G_{x,y}$

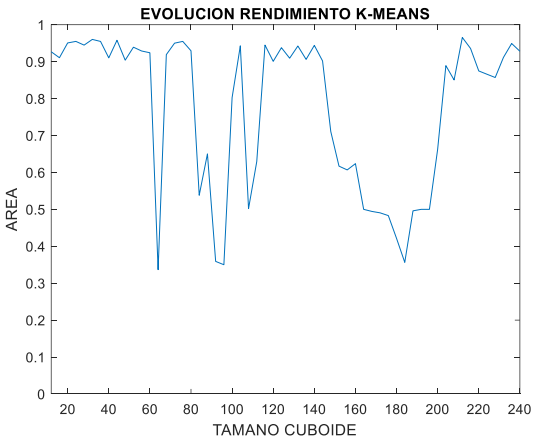


Figura 0-19: Rendimiento UMN 2 con KM y HOG

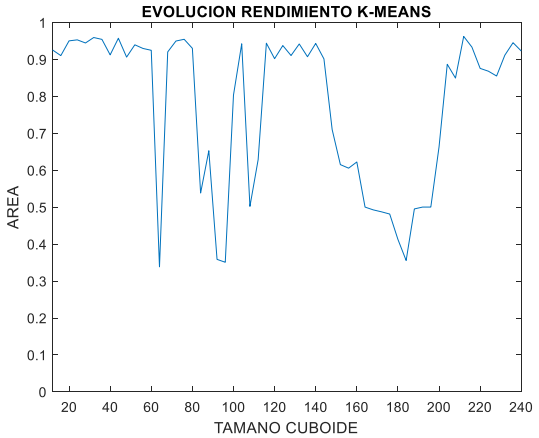


Figura 0-20: Rendimiento UMN 2 con KM y $[G_t, HOG]$

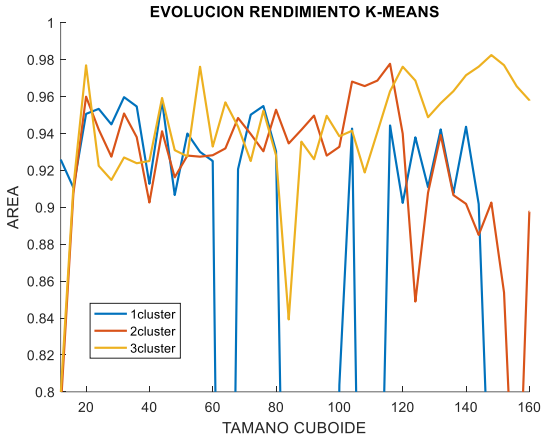


Figura 0-21: Comparativa Clusters UMN 2 con KM y HOG

Tabla 0.4 Rendimiento UMN 2 con FCM

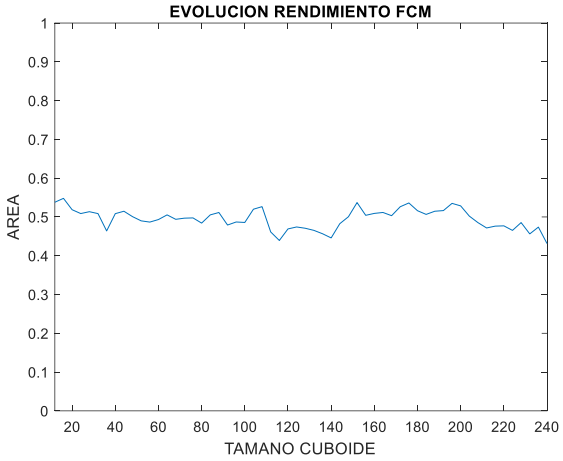


Figura 0-22: Rendimiento UMN 2 con FCM y G_t

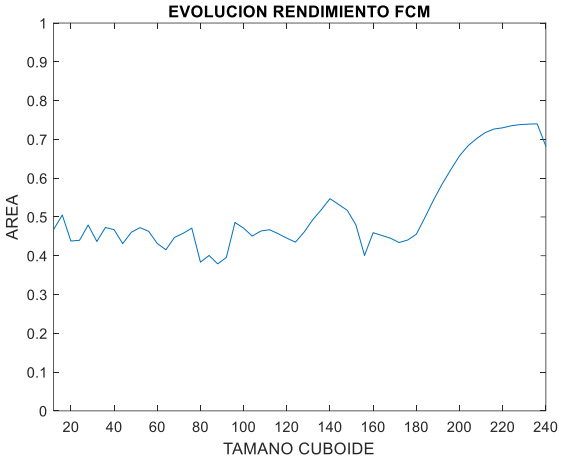


Figura 0-23: Rendimiento UMN 2 con FCM y $ZCR_{x,y}$

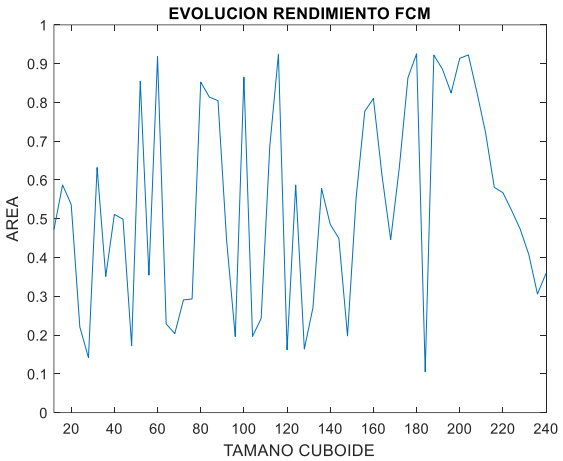


Figura 0-24: Rendimiento UMN 2 con FCM y $G_{x,y}$

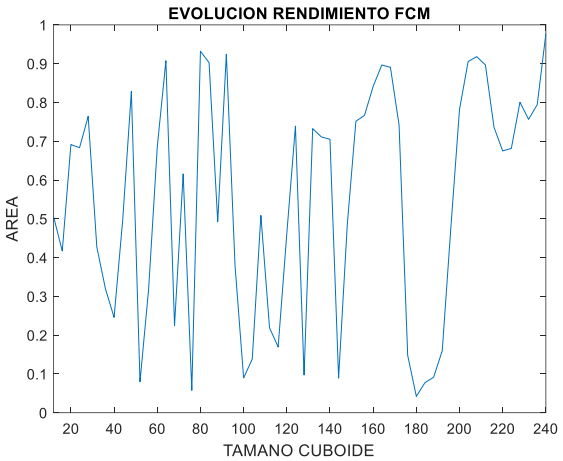


Figura 0-25: Rendimiento UMN 2 con FCM y HOG

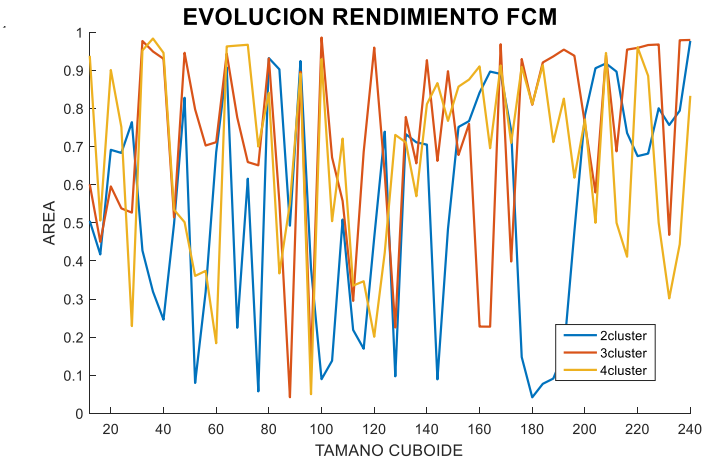


Figura 0-26: Comparativa Clusters UMN 2 con FCM y [HOG, Gt]

Tabla 0.5 Rendimiento UMN 3 con KM

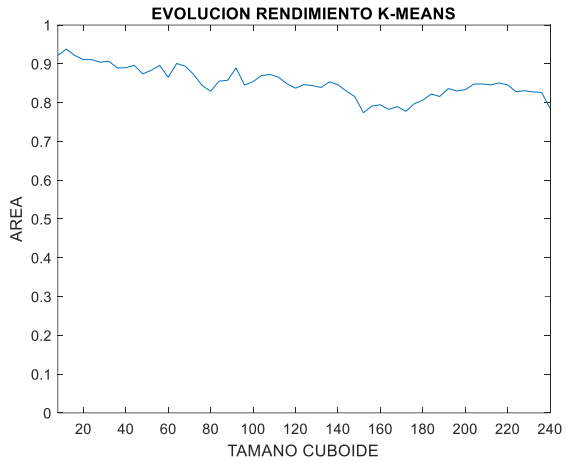


Figura 0-27: Rendimiento UMN 3 con KM y G_t

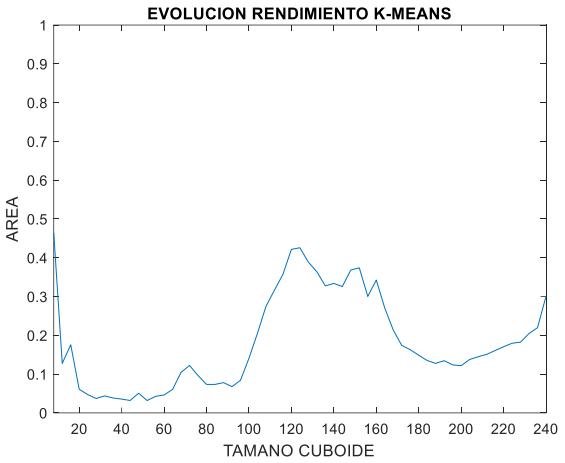


Figura 0-28: Rendimiento UMN 3 con KM y $ZCR_{x,y}$

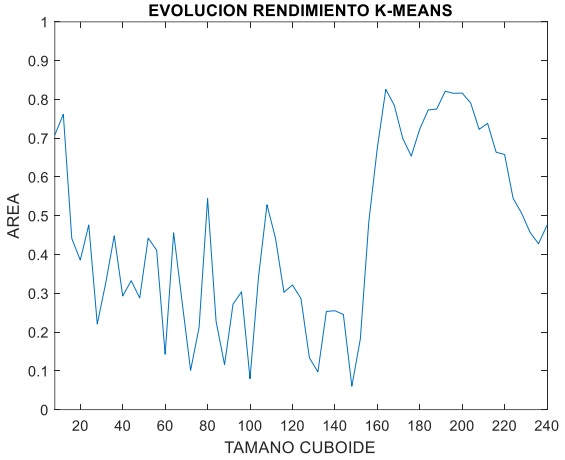


Figura 0-29: Rendimiento UMN 3 con KM y $G_{x,y}$

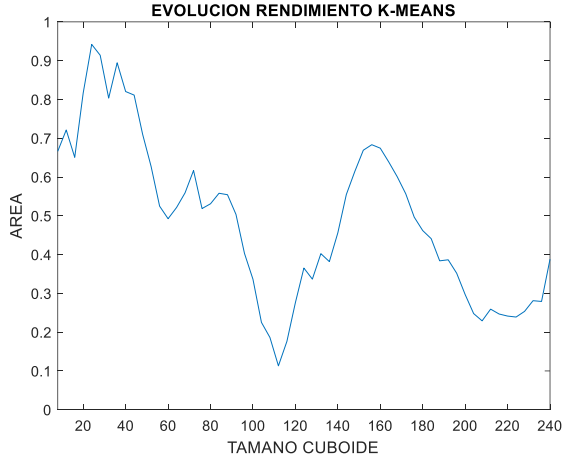


Figura 0-30: Rendimiento UMN 3 con KM y HOG

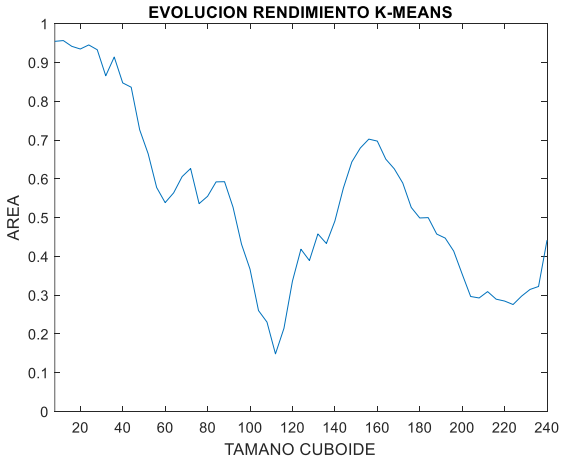


Figura 0-31: Rendimiento UMN 3 con KM y $[G_t, HOG]$

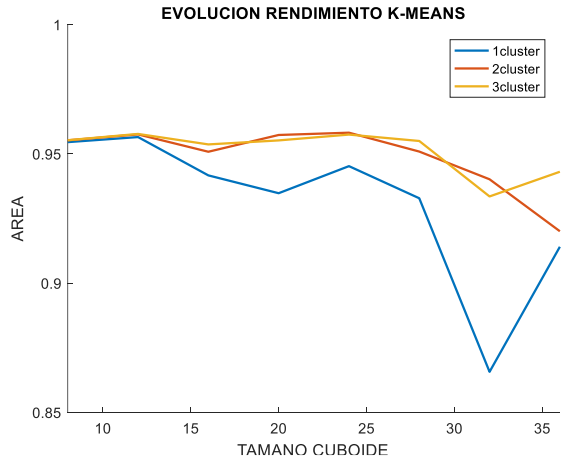


Figura 0-32: Comparativa Clusters UMN 3 con KM y $[G_t, HOG]$

Tabla 0.6 Rendimiento UMN 3 con FCM

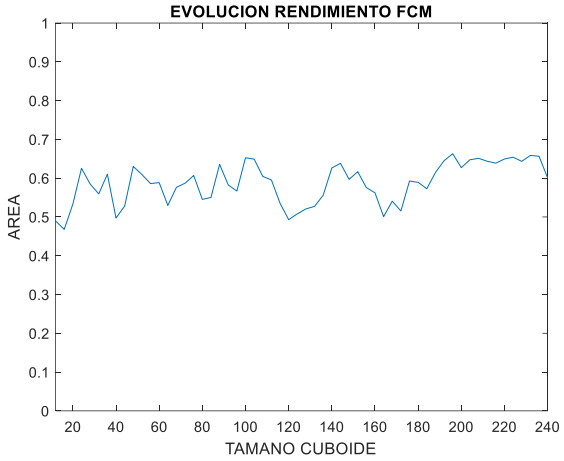


Figura 0-33: Rendimiento UMN 3 con FCM y G_t

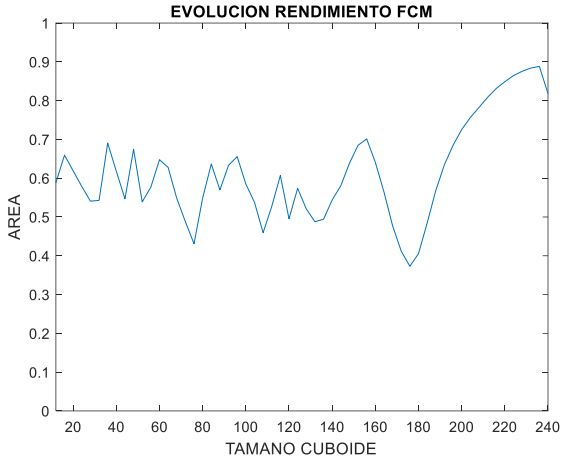


Figura 0-34: Rendimiento UMN 3 con FCM y $ZCR_{x,y}$ y comparativa Clusters

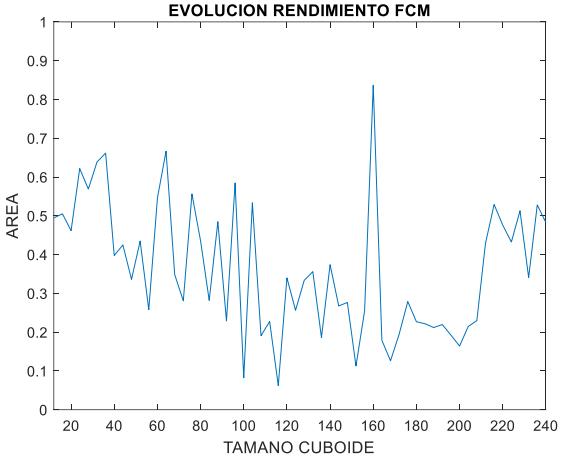


Figura 0-35: Rendimiento UMN 3 con FCM y $G_{x,y}$ y comparativa Clusters

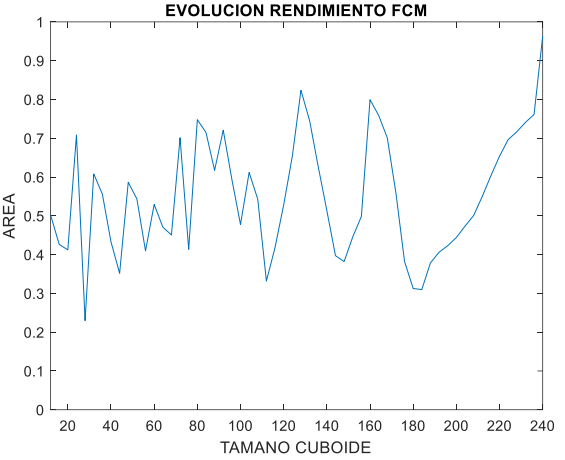


Figura 0-36: Rendimiento UMN 3 con FCM y HOG y comparativa Clusters

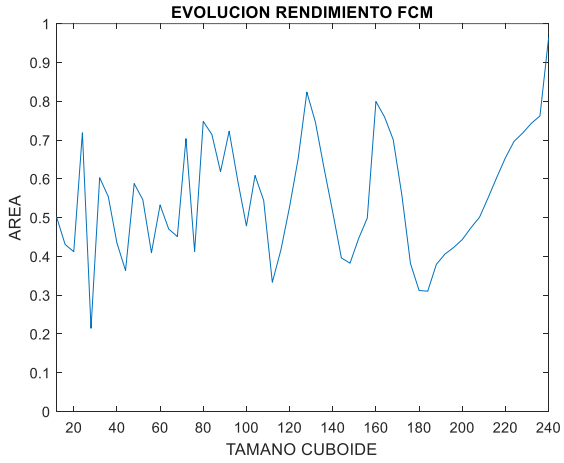


Figura 0-37: Rendimiento UMN 3 con FCM y $[ZCR_{x,y}, HOG]$

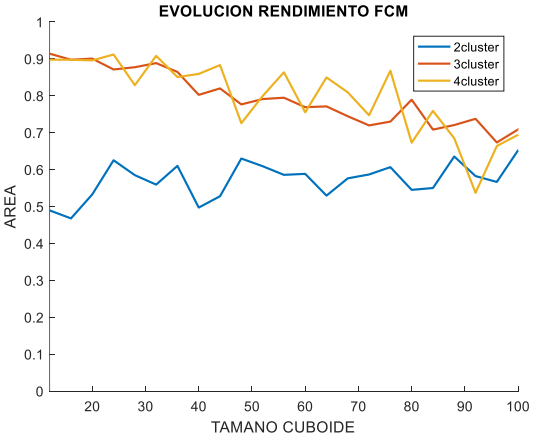


Figura 0-38: Comparativa Clusters UMN 3 con FCM y $[Gt, HOG]$

Tabla 0.7 Rendimiento *anomalous_run_left* con KM

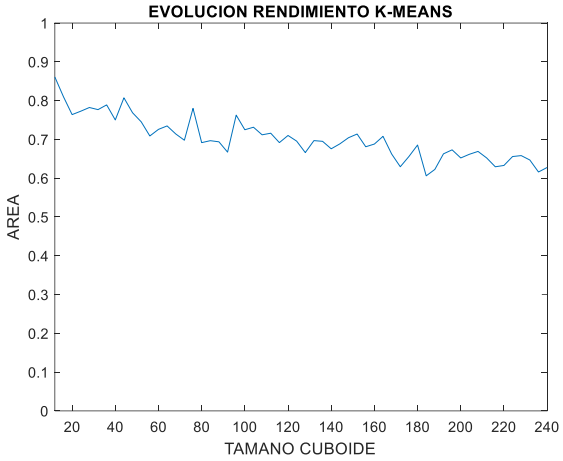


Figura 0-39: Rendimiento *anomalous_run_left* con KM y G_t

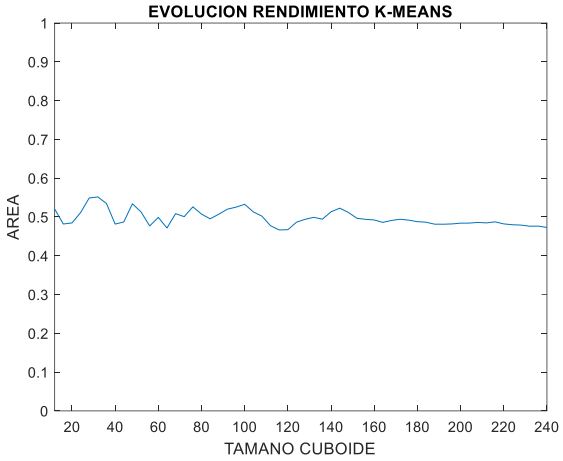


Figura 0-40: Rendimiento *anomalous_run_left* con KM y $ZCR_{x,y}$

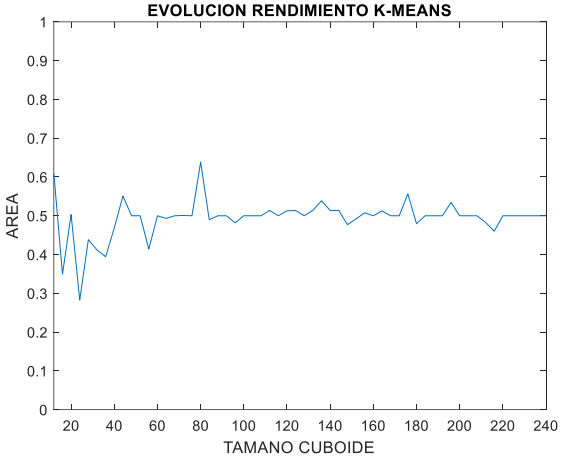


Figura 0-41: Rendimiento *anomalous_run_left* con KM y $G_{x,y}$

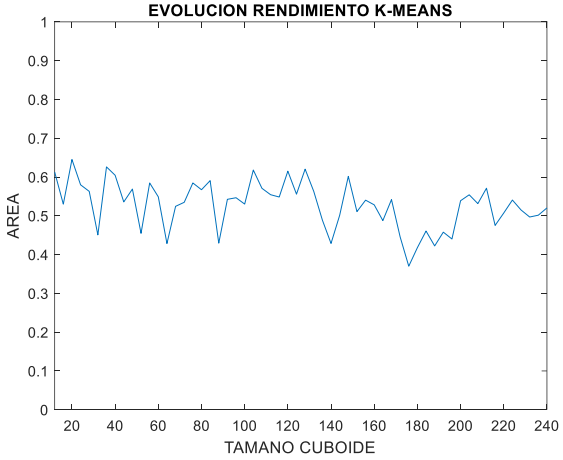


Figura 0-42: Rendimiento *anomalous_run_left* con KM y HOG

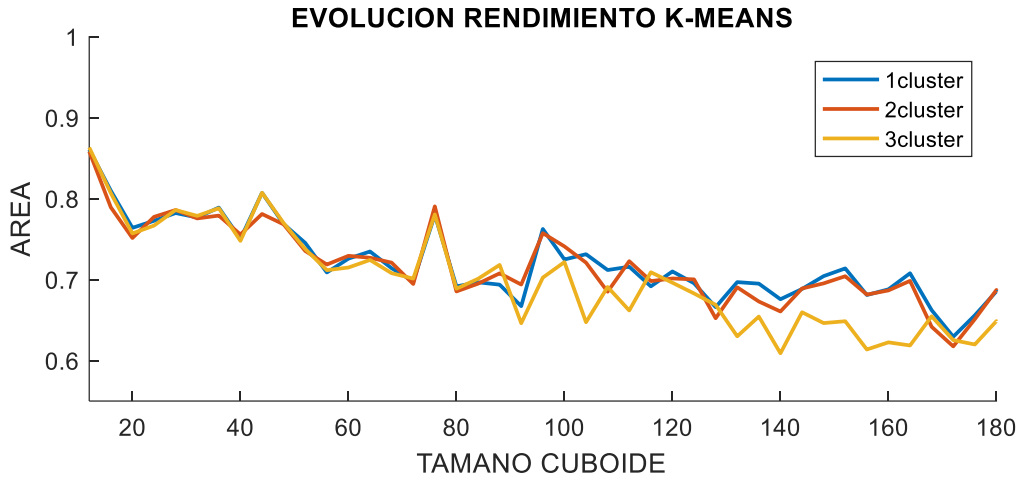


Figura 0-43: Comparativa Clusters *anomalous_run_left* con KM y G_t

Tabla 0.8 Rendimiento *anomalous_run_left* con FCM

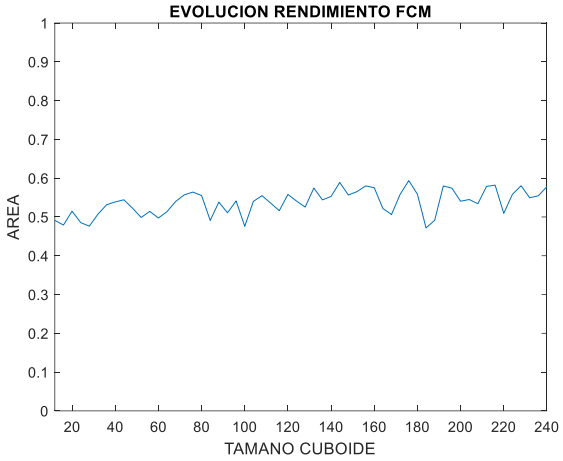


Figura 0-44: Rendimiento *anomalous_run_left* con FCM y G_t

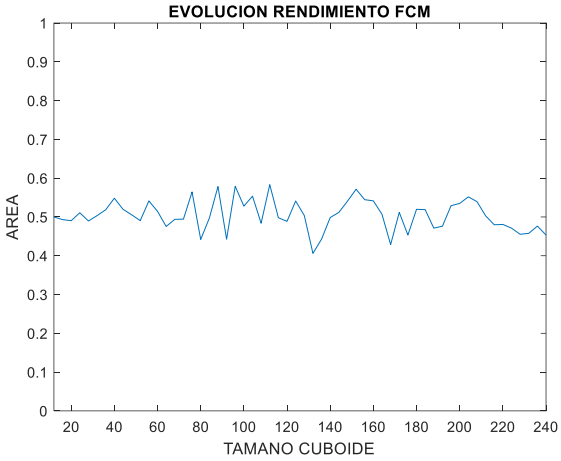


Figura 0-45: Rendimiento *anomalous_run_left* con FCM y $ZCR_{x,y}$

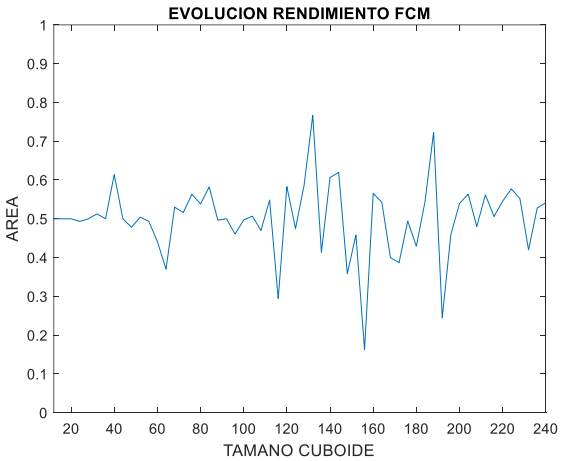


Figura 0-46: Rendimiento *anomalous_run_left* con FCM y $G_{x,y}$

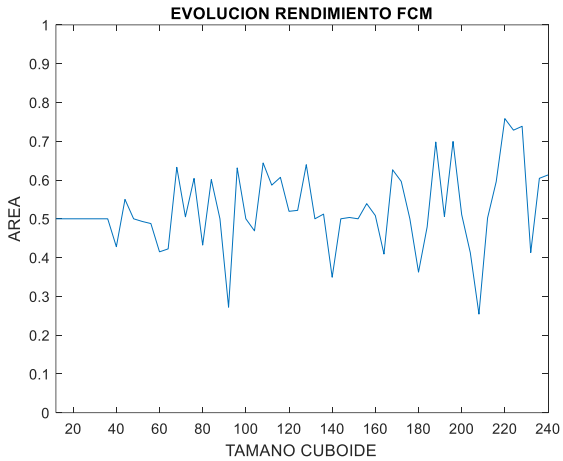


Figura 0-47: Rendimiento *anomalous_run_left* con FCM y HOG

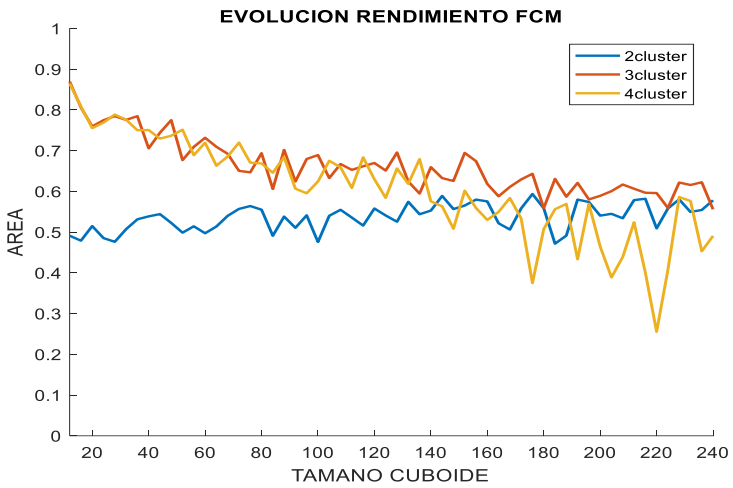


Figura 0-48: Comparativa Clusters *anomalous_run_left* con FCM y G_t

Tabla 0.9 Rendimiento *anomalous_push_left* con KM

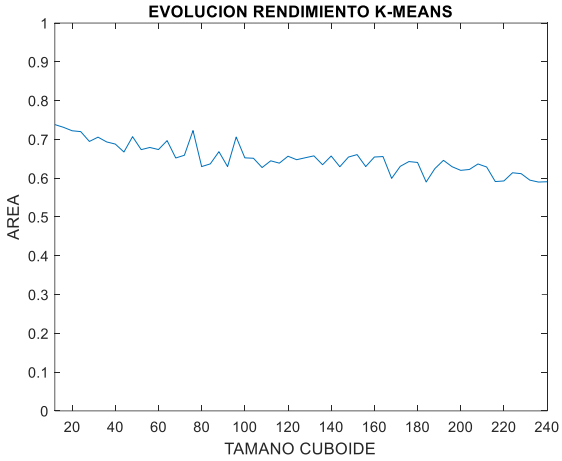


Figura 0-49: Rendimiento *anomalous_push_left* con KM y G_t

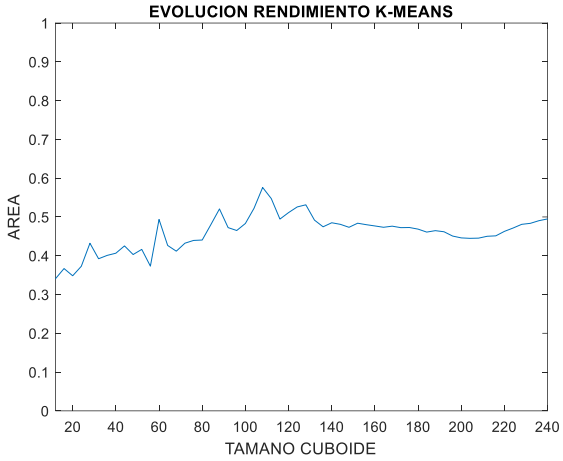


Figura 0-50: Rendimiento *anomalous_push_left* con KM y $ZCR_{x,y}$

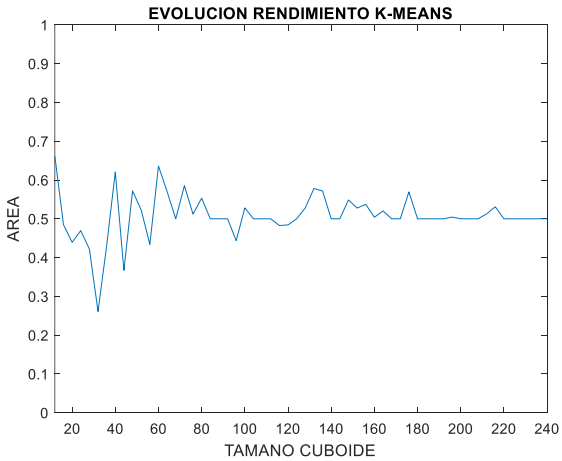


Figura 0-51: Rendimiento *anomalous_push_left* con KM y $G_{x,y}$

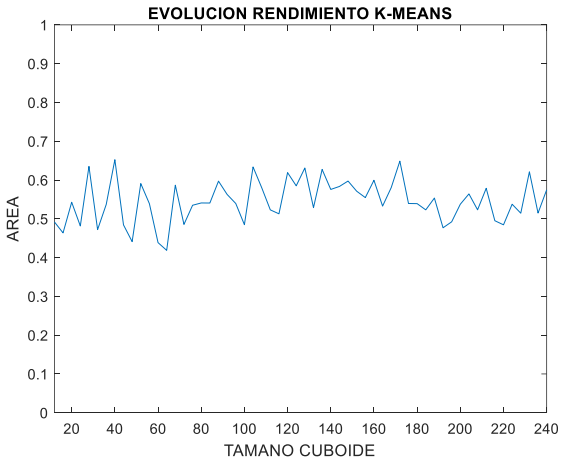


Figura 0-52: Rendimiento *anomalous_push_left* con KM y HOG

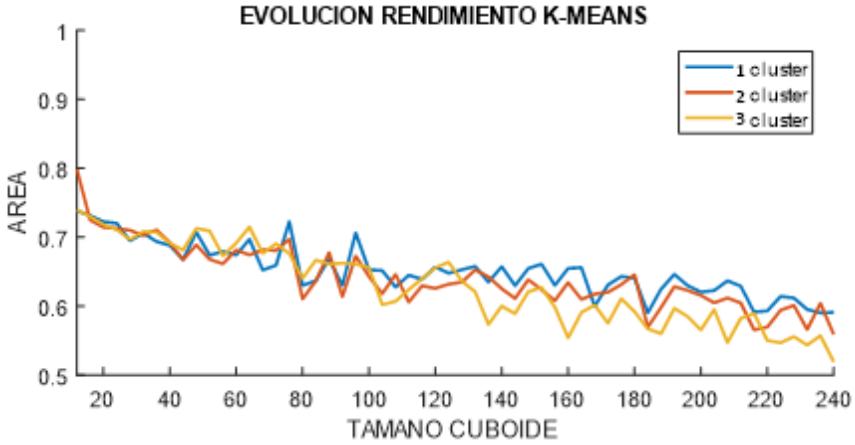


Figura 0-53: Comparativa Clusters *anomalous_push_left* con KM y Gt

Tabla 0.10 Rendimiento *anomalous_push_left* con FCM

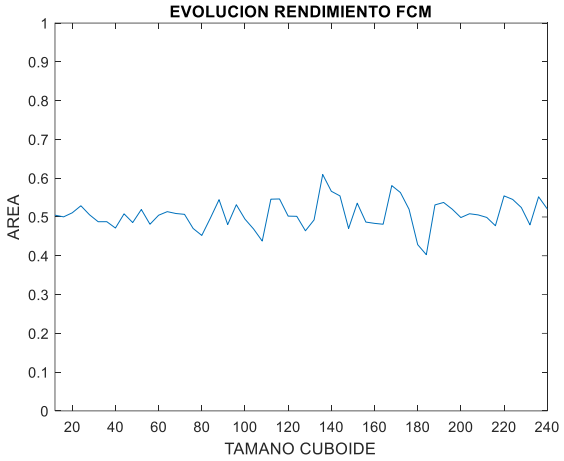


Figura 0-54: Rendimiento *anomalous_push_left* con FCM y G_t

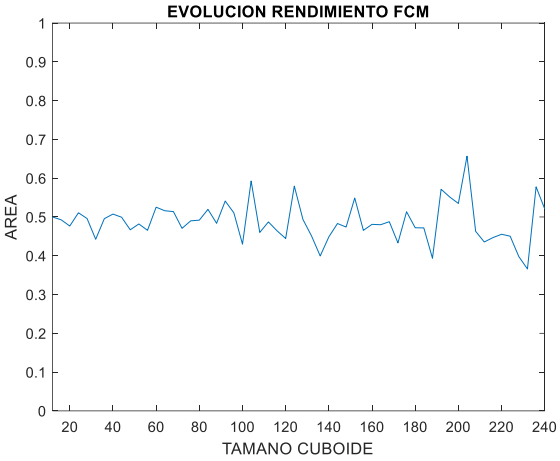


Figura 0-55: Rendimiento *anomalous_push_left* con FCM y $ZCR_{x,y}$

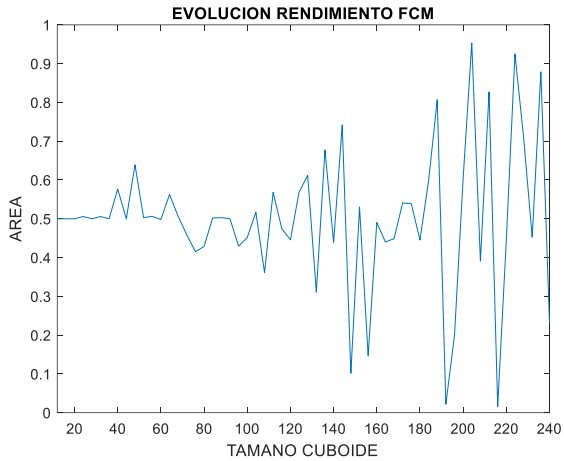


Figura 0-56: Rendimiento *anomalous_push_left* con FCM y $G_{x,y}$

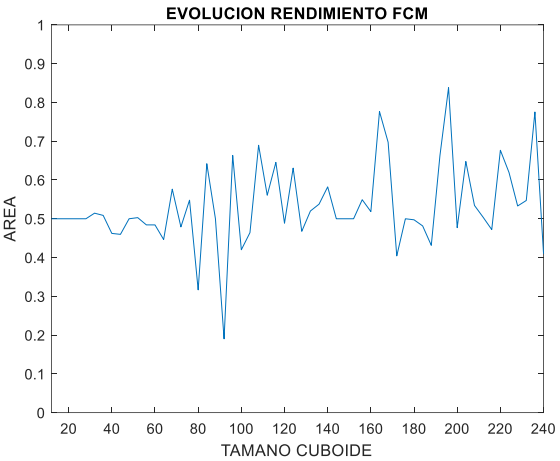


Figura 0-57: Rendimiento *anomalous_push_left* con FCM y HOG

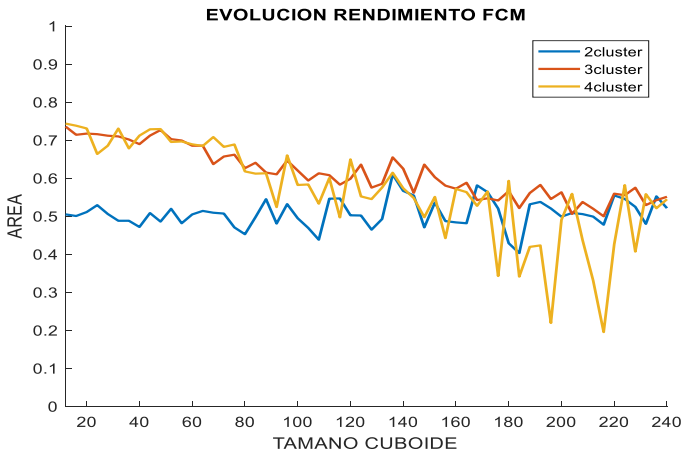


Figura 0-58: Comparativa Clusters *anomalous_push_left* con FCM y Gt

Tabla 0.11 Rendimiento *anomalous_push_up* con KM

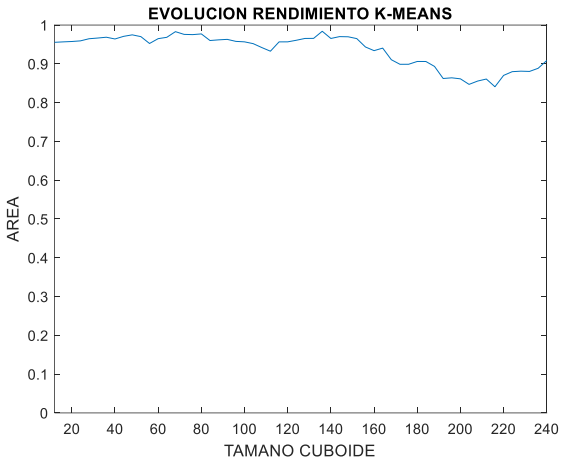


Figura 0-59: Rendimiento *anomalous_push_up* con KM y G_t

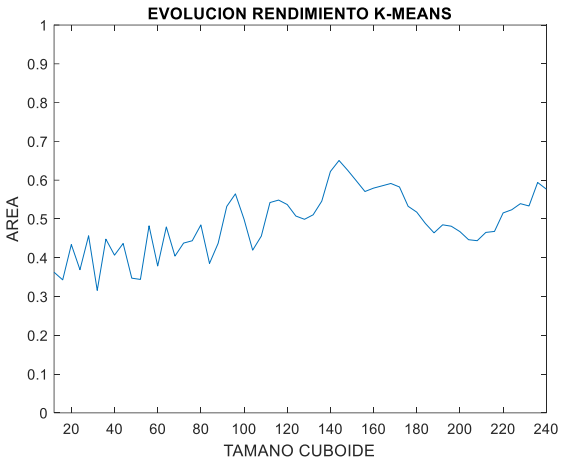


Figura 0-60: Rendimiento *anomalous_push_up* con KM y $ZCR_{x,y}$

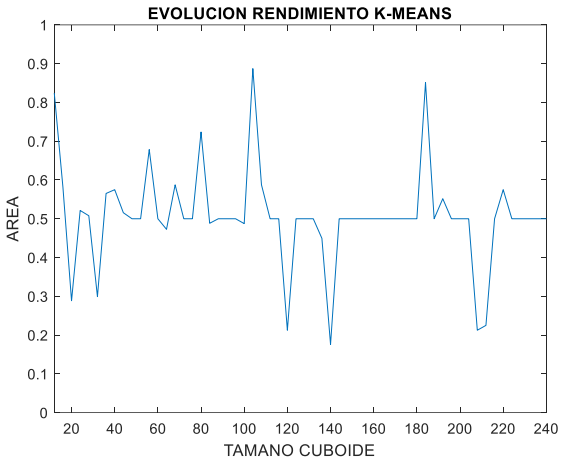


Figura 0-61: Rendimiento *anomalous_push_up* con KM y $G_{x,y}$

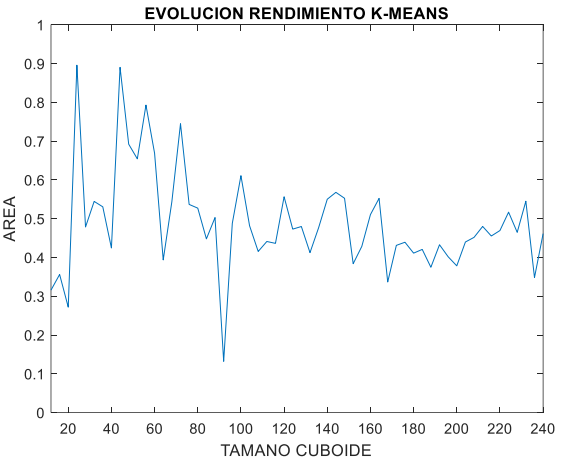


Figura 0-62: Rendimiento *anomalous_push_up* con KM y HOG

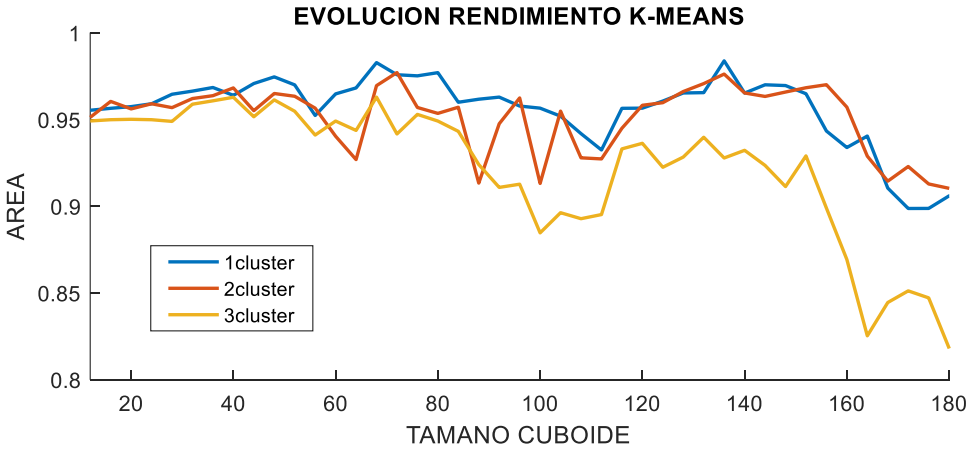


Figura 0-63: Comparativa Clusters *anomalous_push_up* 1 con KM y G_t

Tabla 0.12 Rendimiento *anomalous_push_up* con FCM

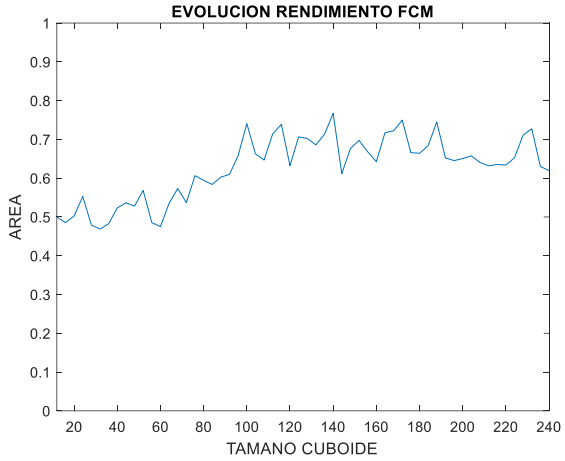


Figura 0-64: Rendimiento *anomalous_push_up* con FCM y G_t

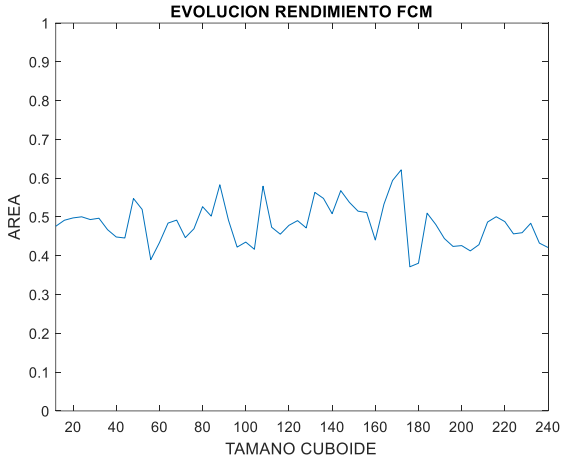


Figura 0-65: Rendimiento *anomalous_push_up* con FCM y $ZCR_{x,y}$

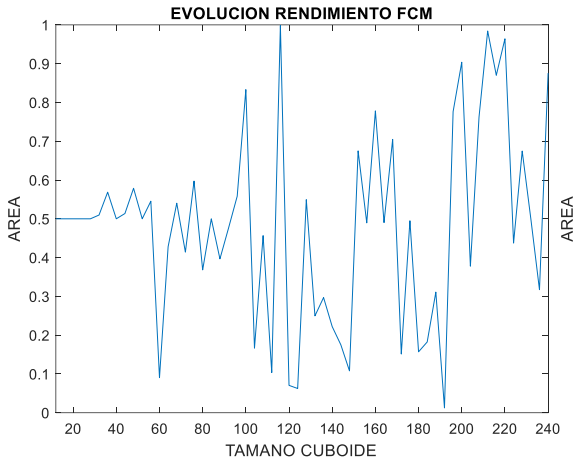


Figura 0-66: Rendimiento *anomalous_push_up* con FCM y $G_{x,y}$

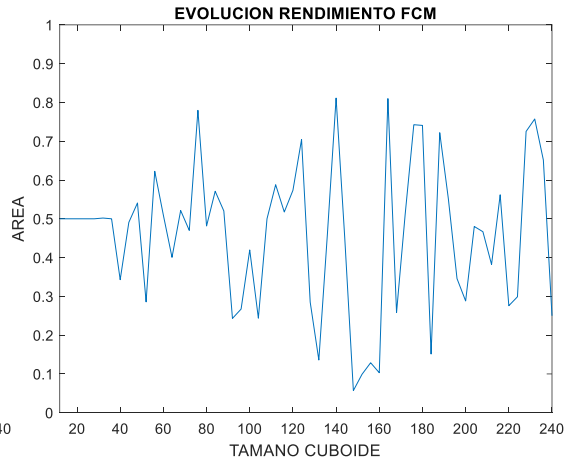


Figura 0-67: Rendimiento *anomalous_push_up* con FCM y HOG

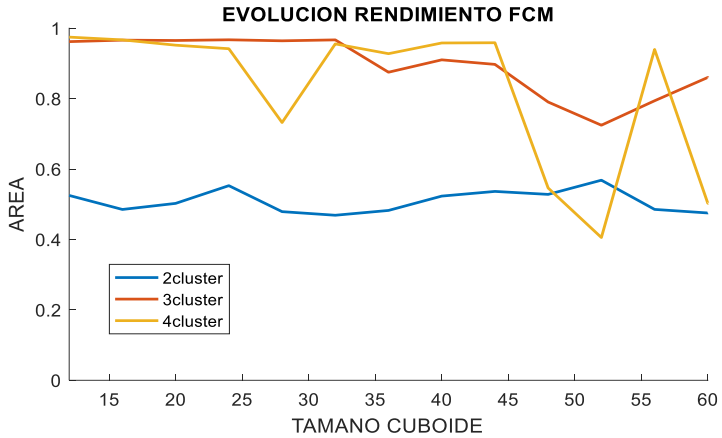


Figura 0-68: Comparativa Clusters *anomalous_push_up* con FCM y G_t

Tabla 0.13 Rendimiento *anomalous_fall_up* con KM

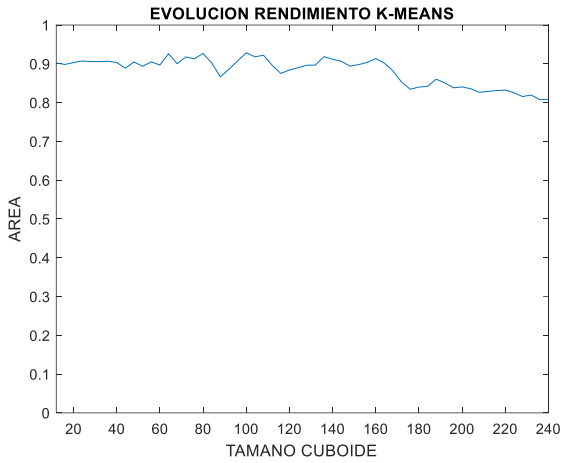


Figura 0-69: Rendimiento *anomalous_fall_up* con KM y G_t

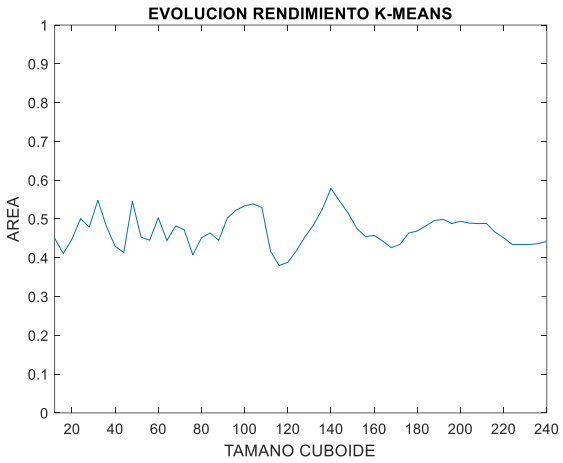


Figura 0-70: Rendimiento *anomalous_fall_up* con KM y $ZCR_{x,y}$

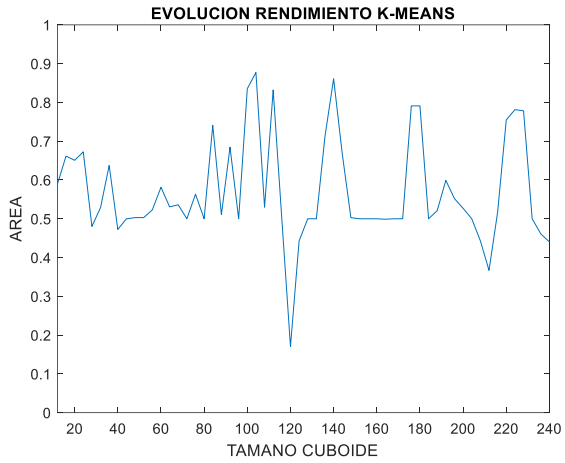


Figura 0-71: Rendimiento *anomalous_fall_up* con KM y $G_{x,y}$

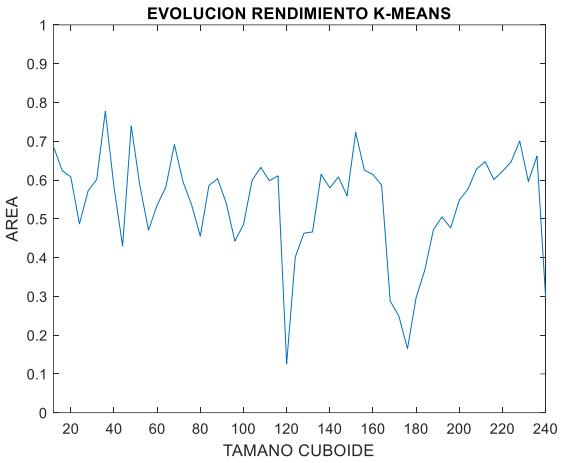


Figura 0-72: Rendimiento *anomalous_fall_up* con KM y HOG

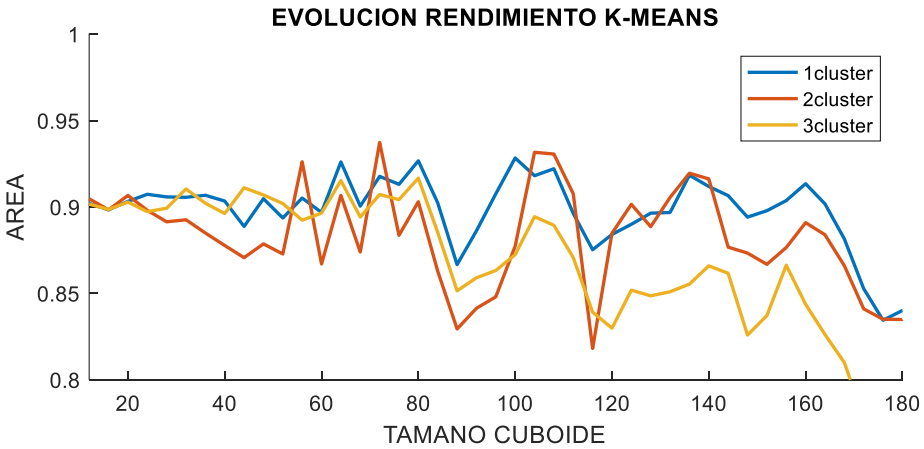


Figura 0-73: Comparativa Clusters *anomalous_fall_up* con KM y Gt

Tabla 0.14 Rendimiento *anomalous_fall_up* con FCM

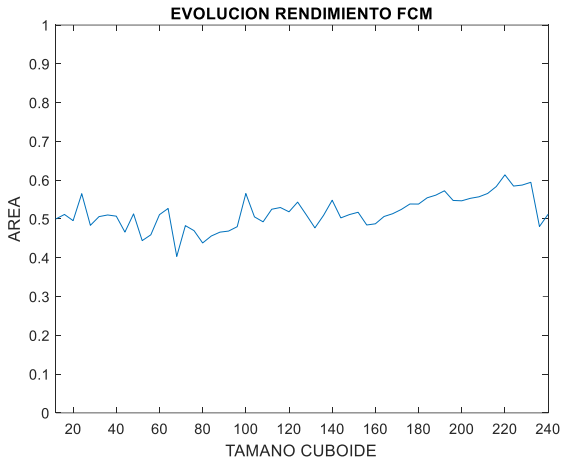


Figura 0-74: Rendimiento *anomalous_fall_up* con FCM y G_t

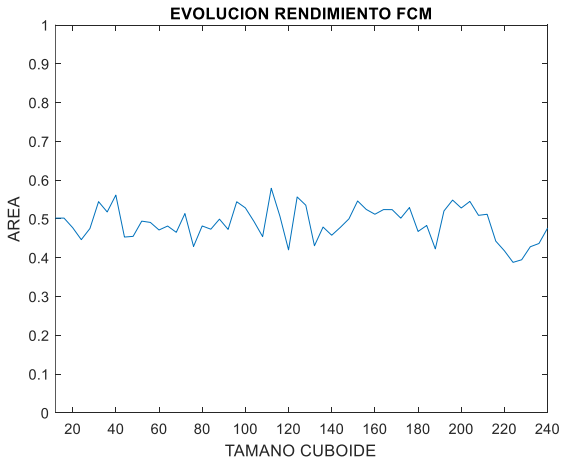


Figura 0-75: Rendimiento *anomalous_fall_up* con FCM y $ZCR_{x,y}$

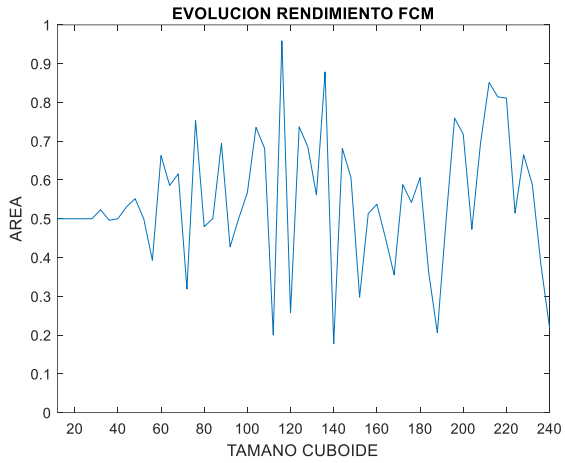


Figura 0-76: Rendimiento *anomalous_fall_up* con FCM y $G_{x,y}$

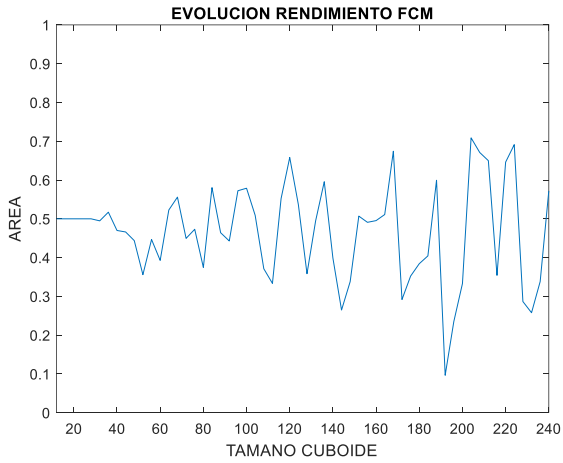


Figura 0-77: Rendimiento *anomalous_fall_up* con FCM y HOG

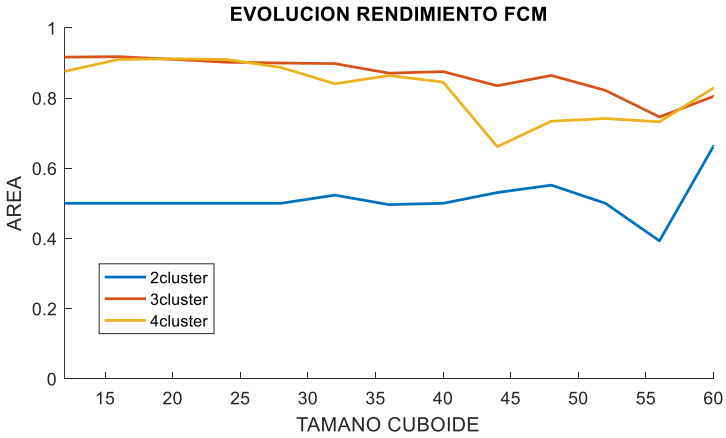


Figura 0-78: Comparativa Clusters *anomalous_fall_up* con FCM y *Gt*

B K-Means/ K-Means ++

KM, también conocido como el algoritmo de Lloyd, es un algoritmo de *hard clustering*, es decir, un algoritmo que asigna cada objeto a un único *cluster* o partición.

El algoritmo calcula iterativamente centroides para cada medida de distancia para minimizar la suma con respecto a la medida especificada. El objetivo, por tanto, de KM, es minimizar la función objetivo del error cuadrático, *squared error* (16)

$$J_{KM}(X;V) = \sum_{i=1}^{c}\sum_{j=1}^{n} {D_{ij}}^2 \qquad (13)$$

donde ${D_{ij}}^2$ es la distancia elegida (normalmente euclídea, consultar Anexo 0), c el número de *clusters* y n el número de puntos en cada *cluster*; X es el conjunto de datos y V las coordenadas de los centroides. De esta manera, los pasos que lleva el algoritmo son los siguientes, moviendo iterativamente los centroides hasta que se alcanza la distancia mínima.

1) Inicialización aleatoria de los *c clusters*.

2) Cálculo de las distacias desde los puntos a los centros de todos los centroides.

3) Asignación de cada punto al *cluster* más cercano.

4) Actualización de centroides siguiendo la fórmula (17).

5) Recálculo de las distancias a los nuevos centroides.

6) Si ningún dato cambia su asignación a los *clusters* se detiene el algoritmo. Si no, se vuelve al paso 3 hasta que se cumpla la condición de parada.

$$v_i = \sum_{i=1}^{n_i} \frac{x_{ij}}{n_i} \;\; ; \qquad 1 \leq i \leq c \qquad (14)$$

Por su parte, KM++ es una modificación del algoritmo que afecta a la inicialización de los centroides de los *cñusters*. Gracias a esta evolución, se puede conseguir optimizar el proceso iterativo reduciendo la complejidad computacional hasta *O(log k)* sin comprometer su precisión. Para ello, se sustituye el paso 1) de la inicialización por los siguientes pasos:

1.a) Se elige un centro V_1 aleatoriamente de las muestras de X.

1.b) Se elige un nuevo centro V_i, seleccionando $x \in X$ con probabilidad $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$.

1.c) Se repite el paso 1.b hasta que se han tomado c centros.

Con esto, se consigue reducir el número posterior de iteraciones de cálculo de distancias y actualización de centroides y se logra mitigar en cierta medida una de las principales limitaciones de KM.

C Fuzzy C-Means

FCM es un algoritmo de *soft clustering, fuzzy clustering* o clasificación suave o dispersa, que asigna cada punto del conjunto de datos a varios *clusters* mediante grados de pertenencia a cada uno, con variaciones desde 0 hasta 1 (frente a algoritmos *hard* como KM que solo contemplan pertenencias 0 y 1). Por ello, en los casos donde no se pueda asegurar que un dato pertenece únicamente a una clase, especialmente *datasets* que puedan contener ruido y *outliers*, funcionará mejor que otro tipo de algoritmos que no contemplen dispersión.

Definiendo como punto de partida el conjunto de n datos, X, y las coordenadas de los c centroides, V, se establece una medida de distancia ${D_{ij}}^2$ (consultar Anexo 0) que se buscará minimizar de manera similar a como se hace en KM. La diferencia respecto a KM es que en este caso se añade un término que indica el peso de los errores cuadrados (19), definiendo U_{c*n} (18) como la matriz de dispersión extraída de los datos X, formada por los elementos u_{ij} de pertenencia a cada *cluster* de centroide V.

$$U = [u_{ij}] \in [0,1] \; ; \quad 1 \leq i \leq c, \;\; 1 \leq j \leq n \tag{15}$$

$$\sum_{i=1}^{c} u_{ij} = 1 \; ; \quad 1 \leq j \leq n \tag{16}$$

$$J_{FCM} = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m \, {D_{ij}}^2 \tag{17}$$

siendo m el parámetro de dispersión (o exponente de los pesos), cuyo valor es un número real mayor que 1, $m \in [1, \infty)$. A medida que este valor se aproxima a 1 la clasificación tiende a ser 'dura' pero cuando, pero cuando crece ésta se vuelve cada vez más dispersa. No obstante, en la mayoría de aplicaciones se toma este valor como 2. Con todo esto, se ejecuta el algoritmo con los pasos que siguen:

1) Inicializar la matriz de pertenencia U aleatoriamente
2) Calcular los vectores prototipo, v_i (21), que forman el conjunto de centroides V, $V = [v_1, \dots, v_c]$
3) Calcular los grados de pertenencia u_{ij} con (22).
4) Comparar $U^{(t+1)}$ con $U^{(t)}$, donde t es el número de iteración.
5) Si $\left|\left|U^{(t+1)} - U^{(t)}\right|\right| < \varepsilon$ entonces se detiene el algoritmo. Si no, vuelve al paso 2.

$$v_i = \frac{\sum_{j=1}^{n} {u_{ij}}^m \, x_j}{\sum_{j=1}^{n} {u_{ij}}^m} \; ; \qquad 1 \leq i \leq c \tag{18}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{D_{ij}}{D_{kj}}\right)^{2/(m-1)}} \tag{19}$$

D Distancias

Estos son la distancia euclídea, la del coseno, Jaccard, Chevysev, Manhattan y Minkowski, definidas a continuación. Siendo X e Y dos objetos o *sets* de una determinada clase, tomados en su forma vectorial como vectores A y B n-dimensionales:

Distancia euclídea:

$$D_{euclidea} = \sqrt{\sum_{k=1}^{m} (X_{ik} - X_{jk})^2} \tag{20}$$

Es la medida estándar, la distancia geométrica entre dos puntos. Es también la métrica por defecto asumida en el algoritmo *K-Means*.

Distancia del coseno:

$$D_{coseno} = \theta = arcos \frac{A.B}{||A||||B||} \tag{21}$$

Está definida como el ángulo que forman dos vectores n-dimensionales

Distancia Jaccard:

$$D_{jaccard} = J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{22}$$

Esta medida calcula la similitud entre conjuntos mediante sus uniones e intersecciones.

Distancia Manhattan:

$$D_{manhattan} = |X_{ik} - X_{jk}| \tag{23}$$

Distancia Chebyshev:

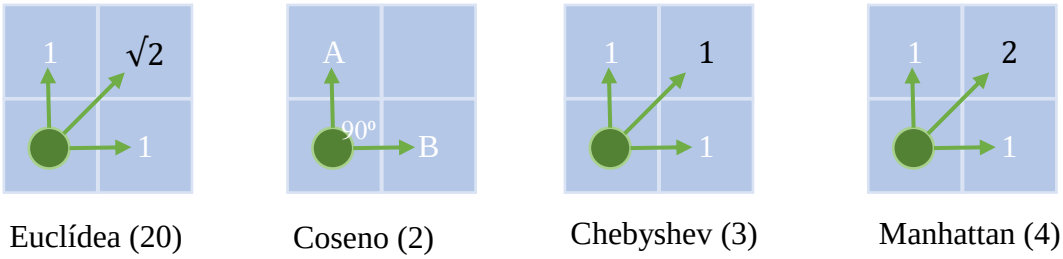
$$D_{chebyshev} = max_k|X_{ik} - X_{jk}| \tag{24}$$

También llamada distancia máxima, calcula la magnitud o valor absoluto de las diferencias entre dos puntos.
Distancia Minkowski:

$$D_{minkowski} = \left(\sum_{k=1}^{d} |X_{ik} - X_{jk}|^{1/p}\right)^p \tag{25}$$

También conocida como la métrica generalizada, pues particularizándola para valores como p=2 se convierte en la distancia euclídea y para p=∞ es la distancia Chevyshev.

De ellas las más reseñables para el tratamiento de imagen son las dos primeras junto con Chevyshev y Manhattan, cuya aplicación a un plano bidimensional puede observarse a continuación en la Figura 0-79:



Euclídea (20) Coseno (2) Chebyshev (3) Manhattan (4)

Figura 0-79: Comparativa distancias euclídea, coseno, Chebyshev y Manhattan 2D